

Alex Malinen

Kaksisuuntaisen sähkömoottorin ohjainkortin suunnittelu ja valmistaminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

DP in Electronics

Insinöörityö

7.5.2017

Tekijä Otsikko	Alex Malinen Kaksisuuntaisen sähkömoottorin ohjainkortin suunnittelu ja valmistaminen
Sivumäärä Aika	33 sivua 7.5.2017
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Elektroniikka
Suuntautumisvaihtoehto	
Ohjaajat	Tekninen johtaja, Reino Malinen Lehtori, Esko Tattari
<p>Insinööritöyssä suunniteltiin ja valmistettiin kaksisuuntaisen sähkömoottorin ohjainkortti. Sähkömoottorin tehtävänä oli toimia hydraulikkapumpun voimanlähteenä. Pumppu oli liitetty kahteen hydraulikkajouseen, joilla levitettiin mekaanisten jousien varassa olevaa testilaitetta. Moottorin kaksisuuntaisuutta käytettiin hydraulikkajousien takaisinvetämiseen. Työn tarkoitus oli osoittaa automaation toimivuus osana hydraulikkajärjestelmää, sekä tuottaa tietoa moottorin vaatimasta virtamäärästä. Tarkoituksena oli myös selvittää automatisointiin, hydraulikkajärjestelmään ja testilaitteen kuormitukseen mahdollisesti liittyviä ongelmia.</p> <p>Ohjainkortti koostui suunnanvaihtopiiristä ja mikro-ohjaimesta. Suunnanvaihtopiirissä transistoreihin kytketyt releet ohjasivat moottorin virtaa ja virranmittausta varten piiriin oli liitetty virta-anturi. Mikro-ohjaimen prosessori luki raja- ja pääkytkinten kautta kulkevia signaaleja ja ohjasi transistoreja saadun tiedon perusteella. Mikro-ohjaimena käytettiin Arduinoa, johon ohjelmoitiin suunnanvaihtopiirin logiikan lisäksi virranmittauksen mahdollistavat funktiot. Arduino oli kytketty kannettavaan tietokoneeseen ja mitatut virrat kirjattiin tekstitiedostoon Gobetwino-ohjelmaa hyödyntäen.</p> <p>Suunnittelussa käytettiin useita simulointi ja suunnitteluohjelmia. Lopullinen ohjainkortti suunniteltiin PADS-ohjelmalla ja tilattiin piirikortteja valmistavalta yritykseltä. Laitteen valmistamisen jälkeen asiakasyritykselle tuotettiin ohjainkortin käyttöön ja kokoamiseen soveltuvat ohjeet, jotka mahdollistavat kortin itsenäisen käytön tulevaisuudessa.</p> <p>Työn aikana selvitettiin useita ongelmia niin automatisoinnin kuin testilaitteen toiminnassa ja niitä hyödynnettiin seuraavan prototyyppilaitteen suunnittelussa. Työhön suunniteltua ohjainkorttia tullaan käyttämään jatkotestauksissa, kun ne ovat ajankohtaisia.</p>	
Avainsanat	elektroniikka, FLX-Frame Oy, tietokoneavusteinen suunnittelu

Authors Title	Alex Malinen Design and Production of Control Circuit for Bidirectional Electric Motor
Number of Pages Date	33 pages 7 May 2017
Degree	Bachelor of Engineering
Degree Programme	Electronics
Specialisation option	
Instructor(s)	Reino Malinen, Technical CEO Esko Tattari, Senior Lecturer
<p>The aim of the thesis work was to design and produce a control circuit for bidirectional electric motor. Motor's purpose was to run a hydraulic pump which was connected to two hydraulic springs. The hydraulic springs were used to expand a test device against mechanical springs. Motors bidirectionality allowed for suppressing the device. The goal of the project was to showcase automation as part of a hydraulic system and to provide information of the current usage of the motor. In addition, the goal was also to find out more about possible problems with the automation process, hydraulic systems and function under stress.</p> <p>Control circuit included a circuit board for changing the direction and a micro controller. The direction changing was implemented in the circuit by a pair of transistors connected to relays. The relays guided the currents way to the motor. The circuit also had an integrated current sensor chip included. The micro controller used signals run through border and main switches to control the transistors of the direction changing circuit. The micro controller used was an Arduino which was programmed with logic for the direction changing circuit and functions for current measurements. Arduino was connected to a laptop and the measured currents were saved to a text file using software called Gobetwino.</p> <p>Several different simulation and design softwares were used during this thesis. The final control circuit was designed with the PADS program from which files were imported to a company able to manufacture the said circuit board. The components for the circuit board were acquired by both the company which requested this thesis work and the designer of the printed circuit board. After completing the control circuit, a set of instructionary guides were provided to the company on using and assembling the circuit which allows independent use of the device in the future.</p> <p>Many problems about using automation and the test device were resolved during this project and information gained was used in the design of the next version of the device. The control circuit designed during this project will be used for further tests in the future.</p>	
Keywords	Electronics, FLX-Frame Oy, Computer aided design

Sisällys

1	Johdanto	1
2	Toteutus	2
2.1	Työn hahmottaminen	3
2.2	Ohjausyksikkö	4
2.2.1	Releet ja transistorit	5
2.2.2	Piirikortin suunnittelu	7
2.2.3	PADS-ohjelma	10
2.2.4	Piirikortin valmistaminen	14
2.3	Datankäsittely	18
2.3.1	Mikro-ohjaimet	18
2.3.2	Arduino	19
2.3.3	Virran mittaus	23
2.3.4	Tiedon keräys ja tallentaminen	23
2.3.5	Moottorin pyörimissuunnan vaihtaminen	24
3	Testaus	26
4	Yhteenveto	28
	Lähteet	32

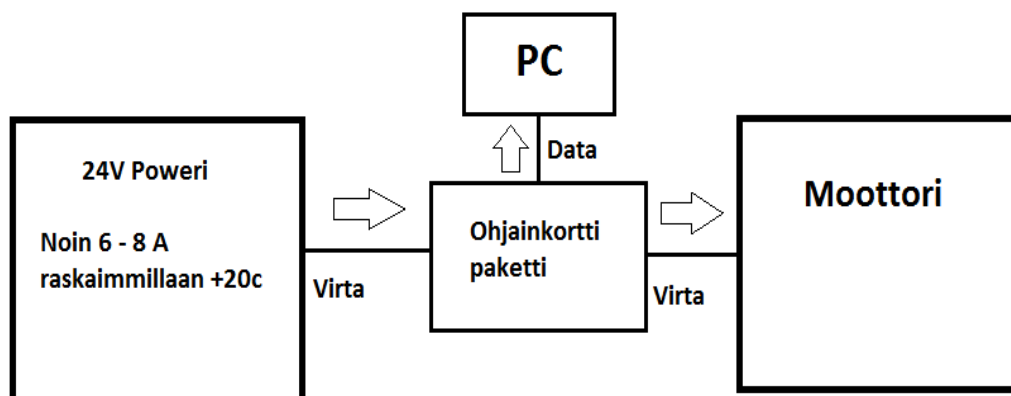
1 Johdanto

Opinnäytetyön tavoitteena on alumiinista ja teräksestä valmistetun hydraulikalla laajennettavan prototyyppielementin testaaminen ja testaukseen käytettävän kaksisuuntaisen sähkömoottorin suunnanohjauspiirin suunnittelu ja toteutus. Moottorin tehtävä on pyörittää hydraulikkapumppua, joka pumppaa painetta prototyyppielementtiä laajentaville hydraulikkajousille. Testauksessa hydraulikkaa ajetaan edestakaisin erilaisissa lämpötiloissa ja kosteusolosuhteissa. Tehtävänä on ajon aikana seurata ja tallentaa virrankulutusta sekä ajoon kulutettua aikaa. Virrankulutuksesta saatavia tietoja hyödynnetään hydraulikkaneiteiden viskositeetin eli juoksevuuden ja hydraulikkaletkujen paksuuksien määrittämisessä. Testauksessa selvitetään myös laitteessa esiintyvän kitkan ja kuorman vaihtelun vaikutus virrankäyttöön. Saaduista tiedoista pyritään saamaan tietoa laitteen jatkokehitystä ajatellen. Myöhemmissä prototyypeissä samankaltaisella systeemillä on tarkoitus levittää laajennettavaa alumiinista ja teräksestä kasattua suurempaa moduulia, joka sijoitettaisiin pakettiauton peräosaan. Työhön ei kuulu erityisten johtopäätösten tai kehitysehdotusten tekeminen testauksen aikana ilmenneistä seikoista vaan yritys hoitaa jatkosuunnittelun omalla asiantuntemuksellaan. Työaiheen minulle tarjosi FLX-Frame Oy.

FLX-Frame Oy on vuonna 2014 perustettu yritys, jonka tarkoitus on kehittää ja kaupallistaa omia innovaatioitaan. Osaksi lopputyönä toteutettu suunnitteluprosessi on osa yrityksen ensimmäistä tuotetta, FLX-1:stä. FLX-Frame Oy on yksityisesti omistettu ja sen toimitusjohtajana toimii Seppo Kalliomäki. Yritys on saanut toimintaansa Tekesin myöntämää rahoitusta ja solminut yhteistyösopimuksen Suomen Puolustusvoimien kanssa. Tekes on Suomen valtion innovaatorahoituskeskus. Toimitusjohtajan lisäksi FLX-Frameella työskentelee vakituisesti tekninen johtaja Reino Malinen ja kaupallinen johtaja Petri Peijari.

2 Toteutus

Kokonaisuudessaan testilaite koostuu lineaarisäätölaitteen mekaniikasta, kevythydrauliikkapumpusta, pumpun sähkömoottorista, hydrauliikkasynteristä ja sen letkuista, sähkömoottorin ohjausyksiköstä, ohjausohjelmasta, datanmuunto-ohjelmasta sekä tietokoneesta. Näistä suunnittelutyöhön kuuluivat moottorin ohjausyksiköksi suunniteltava ja valmistettava piirikortti, ohjausohjelman sisältävä mikro-ohjain sekä datankäsittely mikro-ohjaimen ja tietokoneen välillä. Työ siis muodostui moottorin ohjaamiseen tarkoitettun ohjainkorttipaketin valmistamisesta. Ohjainkorttipaketin tehtävänä oli fyysisesti ohjata moottoria saatujen ohjeiden mukaisesti. Työhön kuului myös testilaitteen aivoina toimivan mikro-ohjaimen liittäminen ohjainkorttiin. Mikro-ohjaimen vastuulla oli eri kytkinsignaalien lukeminen ja niiden vaikutuksesta tehtävät toimenpiteet moottorin ohjauksessa. Mikro-ohjaimella myös lähetettiin kerättyjä mittauksia tietokoneelle myöhempää datankäsittelyä odottamaan (kuva 1).



Kuva 1. Yksinkertaistettu kuva työn sisältämästä kokonaisuudesta. Powerin kohdalle merkitty virtamäärä mitattu moottorin työntäessä jousia vastaan.

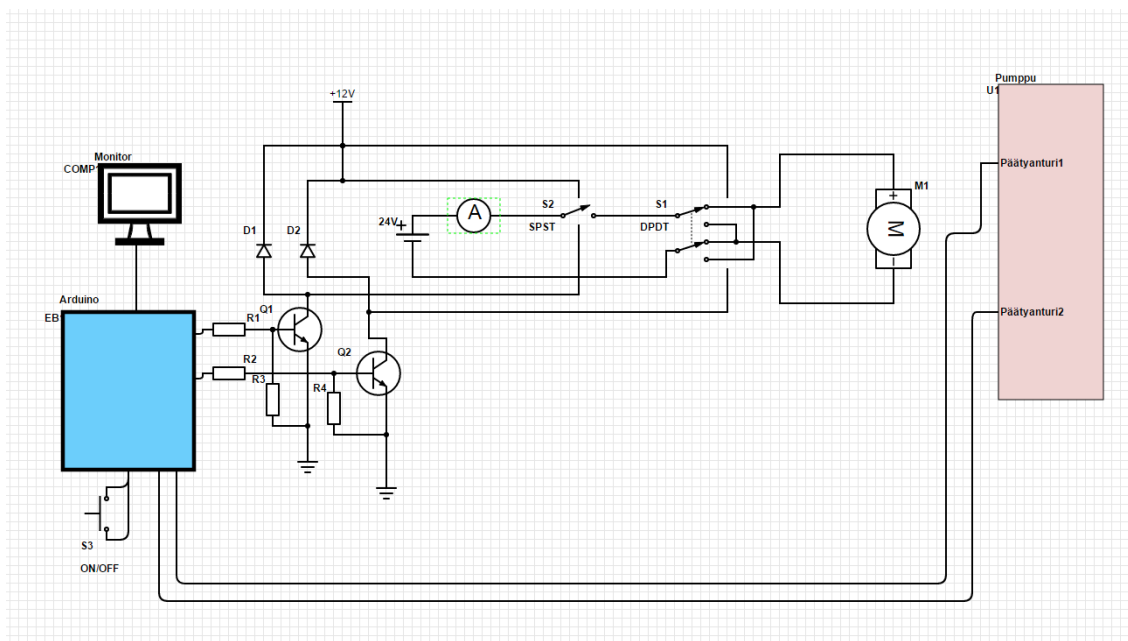
Opinnäytetyön tekeminen aloitettiin virallisesti aloituspalaverissa, jossa käytiin läpi FLX-Framen tarjoaman työn soveltuvuutta opiskelijalle sekä kuultiin ohjaavan opettajan ajatuksia aiheesta. Kokouksessa ohjaava opettaja selvitti muun muassa, millaisia työkaluja työn toteuttamisessa tulisi käyttää ja kenellä on vastuu työskentelyn eri vaiheista. Yrityksen puolesta ohjaajana toimi tekninen johtaja Reino Malinen ja oppilaitoksen lehtori Esko Tattari.

2.1 Työn hahmottaminen

Työn alussa oli tärkeää selvittää eri tapoja, joilla vaatimusten mukainen laitteisto olisi mahdollista toteuttaa. Sovelsin tähän erinäisiä verkosta löytämiäni ohjelmia, joiden kautta sain kuvan työssä vaadittavista komponenteista ja työvaiheista. Kokonaisuuden hahmottelun lisäksi piirsin useita piirustuksia suunnittelun eri työvaiheista ja esittelin niitä yrityksen johdolle. Tavoitteena oli aina pyrkiä minimoimaan mahdollisten suunnitteluvirheiden määrä konsultoimalla kokeneempia tahoja. Tällä tavoin varmistutaan lopullisen tuotteen laadusta ennen varsinaisen tuotantovaiheen aloittamista.

Simuloinnilla tarkoitetaan todellisuuden jäljittelyä yleensä virtuaalisessa ympäristössä. Tarkoituksena on hahmottaa mahdollisimman monta toiminnan ongelmakohtaa turvallisessa ja hallitussa ympäristössä. Simulointi on mielekästä sen edullisuuden takia sekä pitkän aikavälin kustannusten säästössä. Simulointi toimii myös suunnittelijan työn tukena. Säästöä tulee simuloitujen ongelmien huomioon ottamisella valmiissa laitteessa. Virheiden huomioiminen ja ominaisuuksien rajattu testaaminen pienentää korjaus ja/tai uudelleen suunnittelun kustannuksia. Joidenkin asioiden simulointi, esimerkiksi lento-simulaattori, on myös merkittävästi turvallisempaa kuin oikealla koneella harjoittelu. Simuloinnissa tulee noudattaa samaa huolellisuutta kuin muussakin suunnittelussa, sillä virheellinen simulointi tuottaa myöskin virheellistä tietoa laitteesta ja sen toiminnasta. (Simulointi 2016.)

Hahmottelu on tässä tapauksessa suunniteltavaan piirikorttiin vaadittavien komponenttien määrää ja laatua, sekä mikro-ohjaimen koodin rakennetta selventävän mallin tekoa. Hahmottelua voi harrastaa eri piirto-ohjelmilla (kuva 2) tai vanhanaikaisemmin kynää ja paperia käyttämällä. Tarkoituksena on tuottaa yksinkertaistettu malli valmiista tuotteesta, josta suunnittelija ja projektin muut tekijät näkevät selkeästi projektissa vaaditut komponentit ja niiden toimintatarkoituksen. Malleista on myös ongelmatilanteissa helpompi työstää ohjelmakoodiin liittyviä ratkaisuja kuin pelkästään valmista tuotetta moneine komponentteineen katsomalla. Suunniteltavan laitteen hahmottaminen on simuloinnin ohella tärkeimpiä aloitustoimenpiteitä projektissa. Sen avulla voidaan nopeasti selvittää projektin johdon ja osallistujien kanssa, onko lähdetty oikeaan suuntaan ja millaisia haasteita voidaan olettaa projektin aikana syntyvän.



Kuva 2. Schemeitillä piirretty kytkentäkaavio.

Suunnittelussa käytettiin useita eri työkaluja parantamaan omaa ja yrityksen edustajien käsitystä ohjainkortin vaatimasta osa- ja työmäärästä. Internetistä löytyvien ohjelmien avulla voitiin myös jossain määrin hahmotella ja testata ohjainkortin ohjelman toimintaa ennen varsinaisen ohjelmakoodin kirjoittamista.

2.2 Ohjausyksikkö

Ohjausyksikön tuli kyetä ohjaamaan sen kautta kulkevaa virtaa, joka johdetaan edelleen testilaitteen sähkömoottorille. Kortin tulisi kyetä myös katkaisemaan virta ja vaihtamaan sen kulkusuuntaa, jotta sähkömoottoria saataisiin ohjattua halutulla tavalla niin kuormaa vastaan kuin kuormasta vapautettuna. Virran katkaisulla on tarkoitus saattaa sähkömoottori pysähdyksiin tietyn ajon loppuksi ja tahdittaa ajokerrat tapahtumaan määrätyllä aikavälillä.

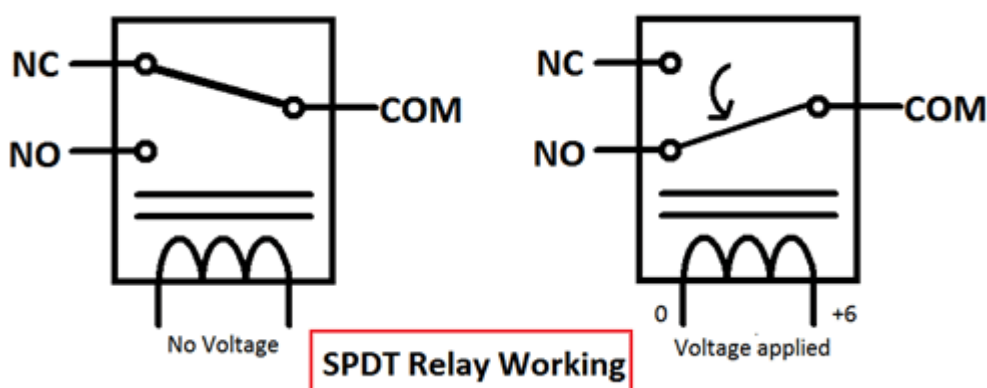
Suunnittelun alkuvaiheessa etsittiin H-silta-komponenttia eli komponenttia jonka sisään ja ulostulojen ristiin kytkentä mahdollistaisi virran suunnanvaihdoksen. H-siltaa ohjataan digitaalisilla signaaleilla esimerkiksi mikro-ohjaimella. Olin havainnollistanut kortin toimintaa aikaisemmin H-sillalla, joten tuntui järkevältä aloittaa siitä. Moottorin suunnanvaihto oli myöskin yksi tärkeimmistä toiminnoista kortissa, joten halusin selvittää sen toimintaa ensimmäisten asioiden joukossa. H-silta olisi yksinkertaistanut korttia jonkin

verran, sillä toisin kuin releet, H-silta ei tarvinnut erikseen transistoria vaan sitä olisi voitu ohjata suoraan mikro-ohjaimen signaaleilla. Moottorin virta oli kuitenkin niin suuri, että sopivaa H-siltaa ei ainakaan järkevään hintaan löytynyt. Etsin siis suhteellisen suurta virtamäärää kestävästä kytkinkomponentista, tai useita sellaisia ja lyhyen pohdinnan jälkeen päädyttiin releisiin, mikä monimutkaisti toteutusta, mutta oli huokeampi ja varmempi toiminnaltaan.

2.2.1 Releet ja transistorit

Releet toimivat kytkiminä ja niitä käytetään transistorien ohella virran ohjaamiseen piirissä. Transistoreissa virta on usein rajattu alle viiteen ampeeriin ja eivät sovi suurempivirtaisten laitteiden ohjainkortteihin. Rele koostuu käämistä sekä erityyppisistä sisään- ja ulostulokoskettimista, joiden lukumäärä ja kytkennät riippuvat releen tyypistä. Rele toimii siten, että käämiin johdettu virta, joka voi olla merkittävästi releen ohjaamaa virtaa heikompi, luo magneettikentän joka liikuttaa releen sisällä olevaa kytkintä kiinni tai auki. Releen käämin vastus ja toimintaan vaadittava jännitemäärä löytyvät kunkin osan ohjetiedostosta. Suurta virtaa ja jännitettä kestävästä releestä nimitetään kontaktoriksi. (Rele 2016.)

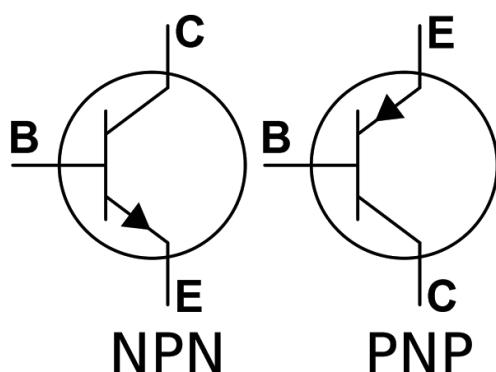
Eri reletyyppejä on useita. Rele voi toimia siten, että virta kulkee Releen koskettimien kautta, kun rele on aktiivinen (normally open) tai virran kulku katkeaa releen aktivoituessa (normally closed). Rele voi myös joko palata lepoasentoonsa virran katketessa tai jäädä viimeisimpään asetukseen. Releen eri kytkentäasetuksia ovat SPST eli single pole single throw, jossa virta kulkee tai ei kulje yhden kytkimen läpi, SPDT (kuva 3) eli single pole double throw, jossa virta kulkee yhdestä liitännästä sisään ja releellä säädetään kytkentää jompaankumpaan ulostuloon, DPST eli double pole single throw, jossa releen läpi kulkee rinnan kaksi SPST-kytkentää ja DPDT eli double pole double throw, jossa rinnan ovat kaksi SPDT-kytkentää.



Kuva 3. SPDT (normally closed) releen toiminta. (Automatic Staircase Lights using PIR Sensor and Relay 2017.)

Transistorit ovat puolijohdekomponentteja, joita voidaan käyttää kytkiminä, vahvistimina sekä muistin elementteinä. Transistorit voidaan jakaa bipolaaritransistoreihin ja kanavatransistoreihin, jotka molemmat hyödyntävät kolmea liitäntää rakenteessaan. Transistorin toimintaperiaatteeseen kuuluu bipolaaritransistorissa kahden liitännän välillä kulkevan virran säätely kolmannella pienemmällä virralla. Kanavatransistoria ohjataan puolijohdekanavassa vaikuttavan sähkökentän avulla. Transistoreita käytetään nykyään käytännössä kaikissa sähköisissä järjestelmissä. (Transistori 2017.)

Bipolaaritransistorin jalat on nimetty kannaksi, kollektoriksi ja emitteriksi. Riippuen siitä, onko kyseessä npn- vai pnp-tyylinen transistori (kuva 4), kannan virralla ohjataan virran kulkua kollektorista emitteriin tai emitteristä kollektoriin. Bipolaaritransistori soveltuu mallista riippuen sekä kauttakulkevan virran vahvistamiseen että kytkimenä toimimiseen eri virroilla. (Transistori 2017.)



Kuva 4. NPN- ja PNP-transistoreiden kytkentäkaaviot. (Transistors 2012.)

Kanavatransistorin jalkoja sanotaan hilaksi, lähteeksi ja nieluksi. Sen toiminta eroaa bipolaaritransistorilla siten, että ohjaus tapahtuu virran sijasta jännitteellä, mutta koko ja rakenne mahdollistavat erilaiset sovellukset. Kanavatransistorit voidaan liittää ohuisiin piikiekkoihin osana integroidun piirin rakennetta. Yhdessä piirissä voi olla miljoonia transistoreita. (Transistori 2017.)

2.2.2 Piirikortin suunnittelu

Kun laitteen sisältämä kokonaisuus ja vaadittavat työvaiheet olivat selvillä, pystyin aloittamaan yksityiskohtiin keskittyvän suunnittelun. Olin suunnittelun alkuvaiheessa käyttänyt Autodesk Circuitsia testilaitteen toiminnan ja ohjelmoinnin hahmotteluun ja lähdin nyt etsimään enemmän laitteen yksityiskohtien suunnitteluun soveltuvaa sovellusta. Tällä kertaa löysin Digikeyn tarjoaman Schemeit -ohjelman, jossa pystyi vapaasti rakentamaan haluamansa piirikaavion, sekä lisäämään omia kytkentäratkaisuja ja muistiinpanoja komponenttien yhteyteen. Schemeitissä ei ollut mahdollisuutta simuloida mikro-ohjaimen ohjelmointia, kuten Autodesk Circuitsissa, mutta uskoin, ettei koodista tulisi yksinkertaisuutensa takia merkittäviä ongelmia. Schemeitissä olisi myös mahdollista valita lisäämilleen komponenteille Digikeyn kaupasta soveltuvat osat ja myöskin tilata osat sisältävä piirilevy valmiiksi kotiinsa. FLX-Frame oli kuitenkin pyytänyt, että osia tilattaisiin yrityksen aikaisemmalta yhteistyökumppanilta, Yleiselektroniikka Oyj:ltä, jolla on jälleenmyyntipiste Suomessa. Digikeyn ainoa varasto sijaitsee Yhdysvalloissa, ja vaikka yritys mainostaa nopealla ja luotettavalla toimitusajalla, olisi toimituksessa voinut kestää kauemmin kuin paikallisesta ketjusta tilattaessa.

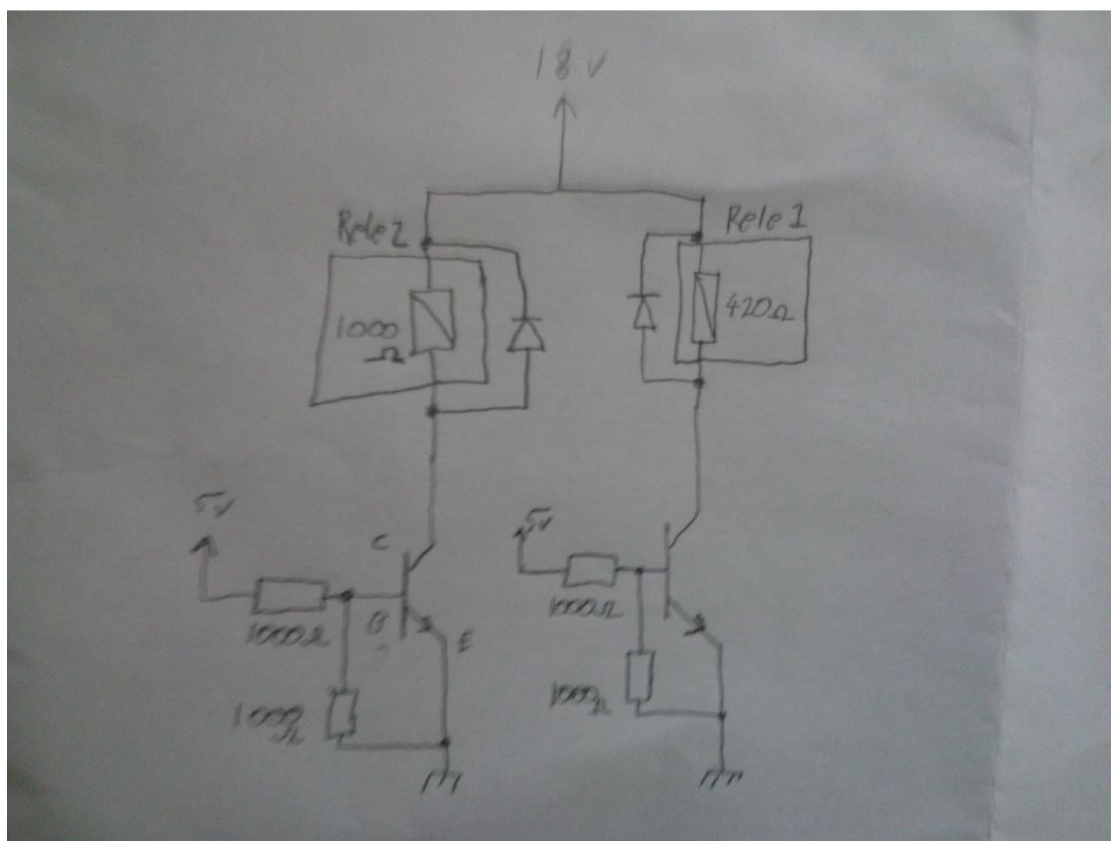
Schemeitillä suunniteltaessa jatkoin aktiivista kanssakäymistä niin firman edustajan kuin ohjaavan opettajan kanssa. Vaikka kaikki komponentit olivat ennestään tuttuja käyttötarkoitukseltaan, oletin jatkuvasti, että kokonaisuudessa saattaisi olla seikkoja, joihin joko koulussa ei olisi tutustuttu tai asiat olisivat päässeet unohtumaan. Teknisten ongelmakohtien lisäksi oli myös tärkeää, että ymmärsin yrityksen vaatimukset laitteen osalta ja siinä auttoi molemminpuolinen aktiivinen kanssakäyminen ja piirrosten esitleminen johdolle. Tapaamisissa oli äärimmäisen tuotteliasta käydä yksinkertaista keskustelua, jossa yrityksen edustaja kysyy, mikä osa ohjainkortissa tekee mitäkin ja miten kortti toteuttaa annetut käskyt. Näin pystyin näyttämään osaamista alasta tai oppimaan uutta, mikäli en pystynyt johonkin kysymykseen vastaamaan. Oli myös hyvä päästä katsomaan

sähkömoottoria ja ajettavaa laitetta (kuva 5) aikaisessa vaiheessa, jotta kävi selväksi, millaisista haasteista ohjainkortin tulisi kyetä selviämään. Piirustusten näyttäminen ohjaavalle opettajalle oli niin ikään tuotteliasta, sillä sen avulla vältin eräiden digitaalisten signaalien kelluvuuteen liittyvän ongelman. Ongelma oli sen laatuinen, ettei sen vaikutusta pystynyt arvioimaan, jollei tuntenut Arduinon ulostulojen toimintaa hyvin. Signaalin kellunnalla tarkoitan signaalia, joka vaihtelee arvoaan, koska sen arvoa ei ole ennestään asetettu ulostuloksi. Arvonvaihtelu johtuu ulkopuolisista häiriösignaaleista, joita Arduino yrittää lukea. Mikäli signaalilla on tarkoitus ohjata moottoria, tulee sen arvo olla hallittavissa.



Kuva 5. Testilaitte, jonka sisällä näkyviä mustia hydraulisia työntöjousia ajetaan edessä näkyvällä pumpulla ja siihen kiinnitetyllä sähkömoottorilla.

Suunnittelutyön edetessä aloitin myös hahmottelemaan piirikorttia PADS-ohjelmalla, josta minulla oli aikaisempaa kokemusta. En ole käyttänyt muita piirikorttisuunnitteluun kykeneviä ohjelmia, mutta oletan, että kaikki vastaavat tietokoneavusteiseen piirisuunnitteluun kykenevät ohjelmat toimivat suurin piirtein samalla periaatteella. Piirikorttisuunnittelussa tulee ensin olla selvä kortin kytkentäkaaviosta (kuva 6) kaikkine komponenttityypeineen, virtalähteineen ja maakytkentöineen. Kun kytkentä on selvä, voidaan ohjelmaan syöttää komponenttien mitat ja jalkojen vaatimat reiänleveydet juottamista varten. Ohjelmassa myös asetetaan komponenttien välillä kulkevien johdinpintojen leveydet ja reitit sekä piirikortin pituus ja leveys. Piirikortin voi suunnitella myös monikerroksiseksi yhden kerroksen sijasta, mikäli kyse on monimutkaisemmasta kortista.



Kuva 6. Piirikortin releosion piirtämäni kytkentäkaavio.

Suunnittelun aikana luin paljon niin Arduinon kuin yleiselektronikan keskustelupalstoja. Monet olivat ennen minua törmänneet omaa laitettaan vaivaaviin ongelmiin, ja ongelmista oli keskusteltu ja ratkaisuja esitetty useita. Keskustelupalstoja seuraamalla onnistuttiin välttämään useita virheitä laitteen suunnittelussa. Esimerkiksi diodien käyttäminen releiden käämien kanssa oli mainittu useissa keskusteluissa. Luettuani, että diodi tarjoaa

turvallisen poistumistien releen käämissä olevalle energialle, kun piirin ohjausvirta katkaistaan, päätin lisätä ne korttiin. Ilman diodeja on mahdollista, että käämissä syntyy suuri jännitepiikki, joka voi vaurioittaa releen yhteyteen lisättyjä komponentteja tai aiheuttaa tarkoituksettoman kytkeytymisen.

2.2.3 PADS-ohjelma

PADS eli Personal Automated Design Solutions on yhdysvaltalaisen Mentor Graphicsin valmistama ohjelma, jota käytetään piirikorttien suunnittelussa. PADS:sillä voidaan suunnitella piirikortin logiikkaa, mittoja sekä myös tehdä 3D-mallinnusta. PADS:in käyttö on suhteellisen helppoa, sillä ohjelman mukana tulee laajat käyttöohjeet ja ohjelma osaa varoittaa automaattisesti monista rakenteellisista virheistä, joita uudelle käyttäjälle voi sattua. PADS:in asetuksista voidaan määrittää erinäisiä minimi- ja maksimiarvoja suunnittelun alla olevaan projektiin jotka vaikuttavat virheentunnistuksen herkkyyteen. Virheentunnistus on myös mahdollista poistaa kokonaan käytöstä, mikäli käyttäjä haluaa suorittaa normaalista käytöstä poikkeavan toimenpiteen. PADS:iin on saatavilla usein valmiita kirjastoja eri komponenteista, joten jokaista osaa ei tarvitset itse luoda esimerkiksi käytettäessä yleisimpiä elektroniikan komponentteja kuten vastuksia ja transistoreita. Aikaa säästyy merkittävästi, kun ohjelmalle ei tarvitse syöttää jokaisen osan mittoja erikseen, vaikkakin on myös hyvä tarkistaa, että kirjastossa olevan komponentin mitat vastaavat käytettävän osan mittoja, mikäli mahdollista. Ohjelmasta näkee myös kätevästi tarkat tiedot siitä, mitä osia on käytetty joka taas helpottaa ja nopeuttaa komponenttien hankintaa. PADS on myös yhdistettävissä partquest.com -sivustoon joka hyödyntää Digi-Key:n laajaa elektronisten komponenttien katalogia osana PADS:in kirjastoa. (PADS Standard 2015: 1—4.)

Kortin valmistusvaihetta edeltävä suunnittelu tehtiin PADS-ohjelmalla. Korttiin sisällytettiin edellä mainittujen releiden, transistoreiden, diodien, vastusten ja releiden lisäksi riviliittimiä Arduinoa ja kytkimiä varten sekä yksi kondensaattori virta-anturin yhteyteen. Kondensaattorin tarkoitus oli parantaa virtamittarin tarkkuutta ja sen käyttöä suositeltiin virtamittarin ohjeissa. Arduino oli kytketty riviliittinten avulla piiriin usean johtimen kautta. Arduinon ohjasi piirissä olevia transistoreita, sekä syötti käyttöjännitteen virta-anturille

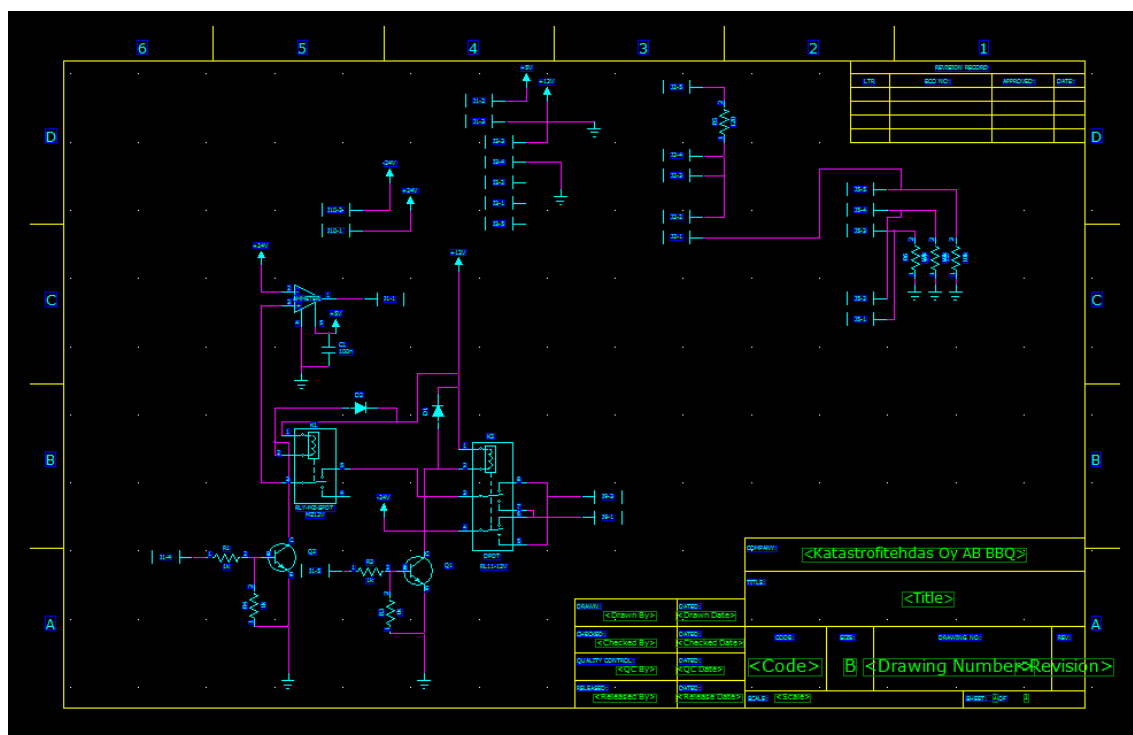
ja kaikille kolmelle kytkimelle. Lisäksi Arduinon nolla oli yhdistetty muun piirikortin nollaan. Päätin myös johdottaa kaikki kytkimet kortin kautta, jotta sain signaalin kulkureitille vastuksia rajoittamaan virtaa.

Piirikortin suunnittelu alkaa piirtämällä PADS logic -osiolla vapaasti kokonaisuus käyttäen joko ohjelman kirjastossa valmiiksi olevia komponentteja, tai mikäli sellaista ei löydy, tekemällä komponentin itse. Tehtäessä komponenttia pitää sille valita tai luoda graafinen symboli niin logic- kuin layout-puolelle. Loogisessa symbolissa on tärkeää sisään- ja ulostulojen määrä joiden pitää vastata osassa olevien määrä, jotta piirikortin johdinreititys saadaan tehtyä. Muuten loogisen symbolin muodolla ei juuri ole väliä lopputuloksen kannalta. Komponentin layout-osion suunnittelussa syötetään ohjelmalle mitat, joiden mukaan piirikorttiin porataan komponentin jaloille sopivat reiät ja liitoskohtaa sivuava johdinpinta. Mitat löytyvät usein komponenttien ohjekirjasesta, tai ne voidaan mitata, mikäli osa on jo saatavilla. Tässä vaaditaan merkittävää huolellisuutta, sillä jo millimetrin kymmenesosan heitto voi aiheuttaa ongelmia joidenkin osien kanssa juotosvaiheessa ja voi johtaa koko kortin uudelleenvalmistukseen. Tärkeää mittojen tarkkuuden lisäksi on varmistaa, että symbolin jalkojen numerointi vastaa komponentin jalkojen numerointia. Näin ohjelma osaa reitityksessä osoittaa mitkä jalat yhdistetään toisiinsa. Esimerkkinä transistorille reikiä asettaessa tulee tietää osan kanta, kollektori ja emitteri.

Kun kaikkien komponenttien mitat ja muodot ovat valmiita, voidaan siirtyä reititykseen. PADS-layoutissa näkyy logicin määrittämät liitoskohdat kaikille komponenteille ja tehtäväksi jää kaikkien reittien piirtäminen ohjelmassa siten että reitit eivät risteä toisiaan aiheuttaen oikosulkuja. Ensimmäiseksi kannattaa lähteä sijoittamaan kukin komponentti erille toisistaan ja mieluiten niin, että samankaltaiset osat ovat suunnilleen samassa nipussa ja yhdistetyt osat olisivat mahdollisimman lähekkäin. Tällä tavalla reititys helpottuu, ja valmiista kortista on helpompi lähteä etsimään mahdollisia vikoja tai korjaamaan vaurioita. Reitityksessä pitää ottaa huomioon myös johdinpinnan leveys ja etäisyys vieressä kulkevaan johdinpintaan. Nämä määräytyvät johtimessa kulkevan virran mukaan, ja mikäli raja-arvot ovat tiedossa, voidaan PADS:ille antaa ohjeet varoittaa liian lähekkäin vedetyistä johdinpinoista. Yleensä reitityksessä pyritään saamaan kortti mahdollisimman pieneen kokoon valmistuskustannusten minimoimiseksi, mutta esimerkiksi prototyyppikortissa koko voi olla toissijainen toimivuuteen ja helppokäyttöisyyteen nähden.

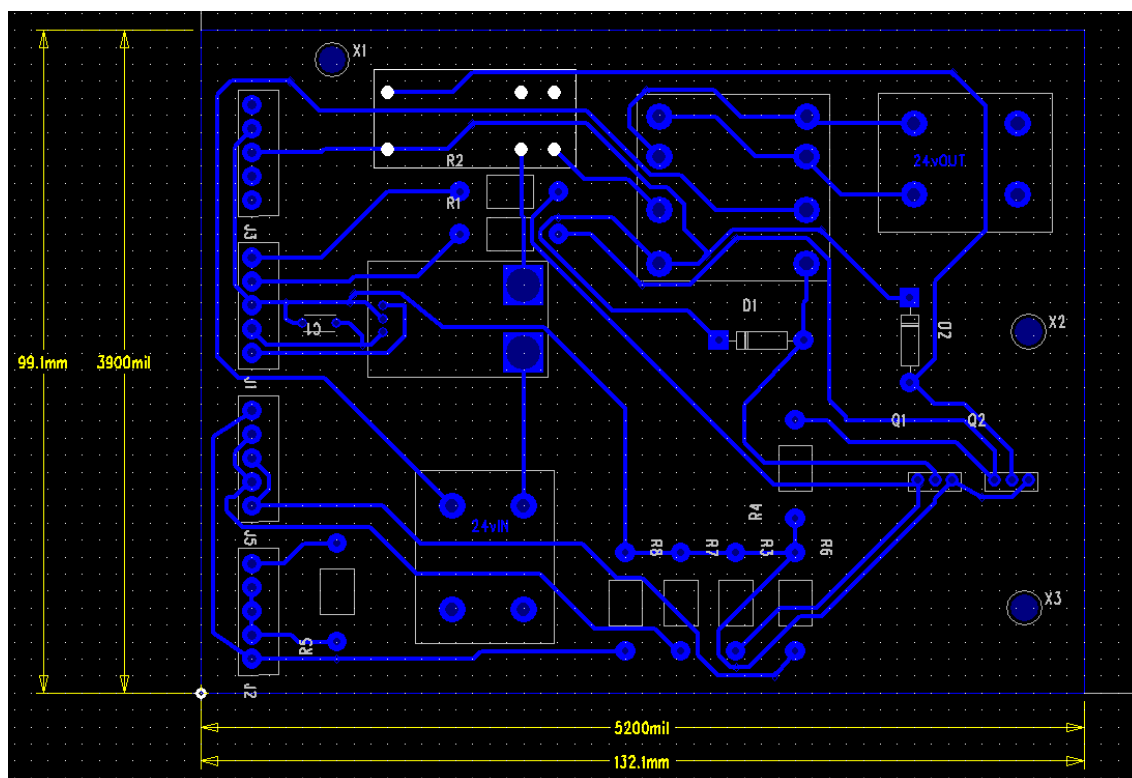
Kun reititykseen ja kortin kokoon ollaan tyytyväisiä, voidaan ohjelmaan lisätä kortin leikkausalue mittoineen sekä kiinnitysreikiä tarpeen mukaan.

Suunnittelu PADS:illa (kuva 7) ei ollut erityisen haastavaa, kiitos koulussa tehtyjen harjoitusten. Se kuitenkin vei aikansa, sillä en aina muistanut, mistä mitäkin asetuksia ohjelmassa säädetään. Päädyin myös tekemään lähes kaikki komponentit itse, sillä loppujen lopuksi niitä ei ollut useita ja pidin mielekkäänä harjoitella niiden tekoa. Vaihtoehtona olisi ollut etsiä valmiita komponenttikirjastoja internetistä, mutta se ei mielestäni olisi säästänyt juurikaan aikaa ja mitat olisi kuitenkin pitänyt tarkistaa hyvän käytännön mukaisesti. Etenin PADS:issa varoen, sillä en halunnut joutua tekemään piirikorttia useaan kertaan, varsinkin kun työstöaikana koulun jyrsimet olivat epäkunnossa ja jouduimme tilaamaan jyrsimen kortin muualta. Olin aikaisemmin jyrsinyt vain yhden pienen kortin koulun koneilla, joten en ollut aivan varma, miten olisin selviytynyt siitä. Muistaakseni jyrsin oli varsin pikkutarkka asetuksistaan ja altis niin käyttäjän kuin omillekin virheilleen.



Kuva 7. Piirikortin valmis logiikka.

Komponenttien mallintamisessa tuli olla huolellinen. Lähes kaikissa komponenteissa oli tarkat mitat merkittynä kunkin ohjevihkoseen, mutta koska minulla oli suunnittelun tässä vaiheessa kaikki tärkeimmät komponentit hankittuna, otin mitat suurimmasta osasta osia itse. Tärkeää oli merkitä oikeankokoiset reiät komponenttien jaloille sekä tarkistaa, että reiät olivat komponentin jalkoja vastaavilla paikoilla. Releiden osalta oli myös tarkistettava, että levyn päällä, eli puolella jossa komponentit olivat, oli tarpeeksi tilaa. Varsinkin suunnanvaihtorele oli muihin osiin verrattuna varsin kookas. En ollut selvillä vielä kaikkien vastusten arvoista, mutta koska vastusten ja muiden pienempien komponenttien mitat ovat usein samankaltaisia, pystyin asettamaan niille paikat piirikorttiin ilman, että olisin hankkinut komponentteja tai niiden mittoja. Muuten pyrin noudattamaan PADS:issa minulle opettuja suunnittelun nyrkkisääntöjä, eli samankaltaisia komponentteja yhteen kasaan ja piirilevyn koon pitäminen mahdollisimman pienenä (kuva 8). Jälkimmäisestä tosin joustin, sillä on mielekkäämpää tehdä prototyyppilaitteeseen tarkoitettu testikortti ehkä tarpeettoman suureksi, mikäli se auttaa havaitsemaan tai havainnollistamaan kortissa tapahtuvaa toimintaa. Testauksen aikana paljastui myös muutamia virheitä kortissa, joita saatiin helpommin korjattua, kun kortissa oli säätämisvaraa.



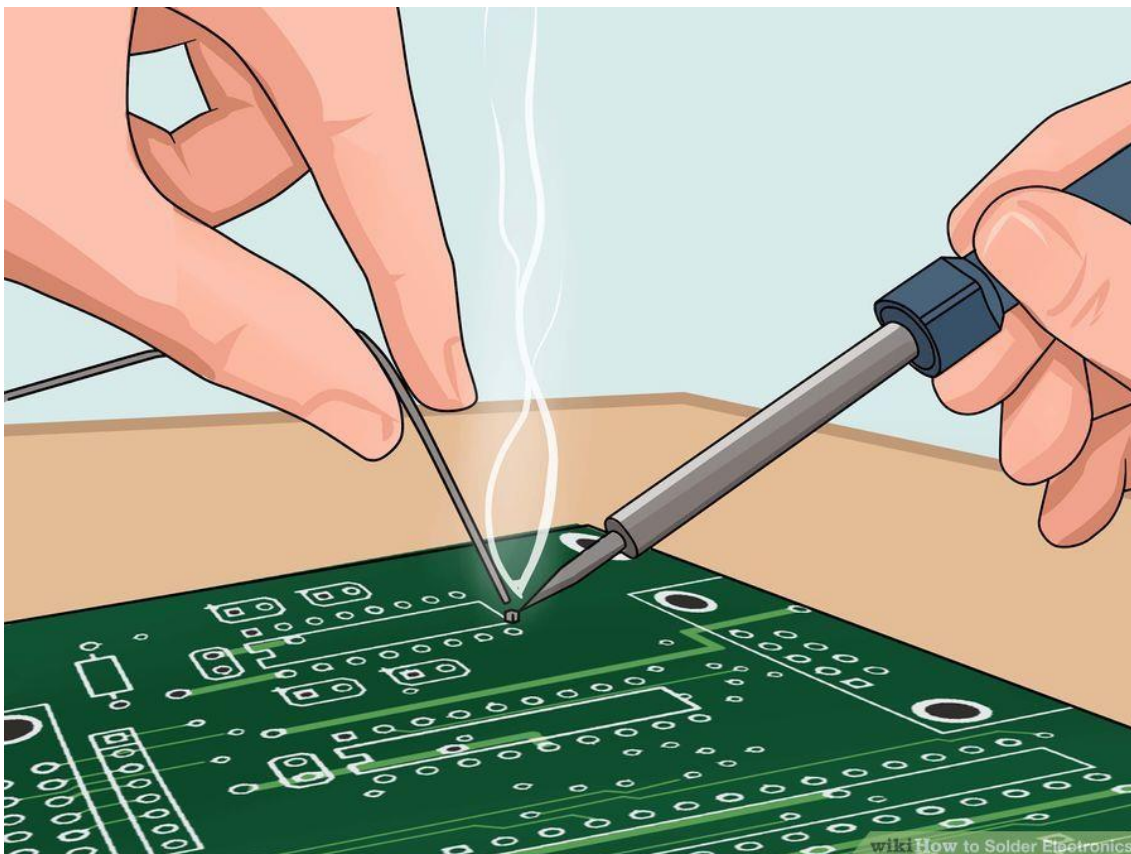
Kuva 8. Piirilevyn jyrksimisohjeet.

2.2.4 Piirikortin valmistaminen

Suunnittelun lopuksi PADS:ssä voidaan luoda jysmiseen vaaditut tiedostot ja viedä ne esimerkiksi CircuitCAMille tai vastaavalle jysmiseen tarkoitettulle ohjelmalle. Ohjelmassa on hyvä tarkistaa, että kortin puolet tai kerrokset ovat kohdistettu toisiinsa ennen jysinohjeiden hyväksyntää. Tämän jälkeen tarkistetaan myös, että jysimessä olevien terien mitat vastaavat kortissa vaadittuja ja terät ovat niille tarkoitetuilla paikoilla. Ennen kortin jysintää jysin tekee piirikortin levymateriaaliin kohdistusreiät, joita käytetään kortin oikeaan kohdistamiseen. Mikäli jysin ei jostain syystä löydä kohdistusreiäitä voidaan ne etsiä manuaalisesti ohjelmiston kautta kameraa liikuttamalla. Tämän työn tapauksessa kortin jysimisen toteutti Prinel Piirilevy Oy joka työsti kortin PADS:ista tuotujen mittojen pohjalta yrityksen tarjoamalle FR4-luokan materiaalista valmistetulle tuotekehitykselle. FR4 on tulenkestävää lasikuituvahvisteista laminaattia, joka on päällystetty epoksilla, eli eräänlaisella kovettuvalla muovilla [FR-4 2017]. Mikäli korttiin on lisätty PADS:in kirjastoista löytyviä komponentteja, on ohjelmalla mahdollista luoda valmis lista kortin sisältämistä komponenteista eli BoM (Bill of Materials). BoM:in avulla kortin suunnittelija tai ulkopuolinen hankkija näkee helposti mitä osia korttiin tulee ja pystyy hankkimaan ne kätevästi.

Joutuimme turvautumaan ulkopuoliseen piirikortin valmistajaan, sillä PADS:illä tehdyn piirilevysuunnittelun jälkeen ja tarvittavien jysmistiedostojen luonnin valmistuttua törmäsin tekniseen vikaan. Kumpikaan koulun jysmistä ei nimittäin toiminut ja tarkkaa tietoa niiden kuntoon tulemisesta ei ollut. Keskustelemalla Metropolian Albertinkadun toimipisteen laboratoriovastaavalle arvioimme, että korjaamiseen menisi muutamia viikkoja. Yritin myös selvittää, olisiko muissa Metropolian pisteissä soveltuvaa jysintä piirilevyn tekemiseen, mutta koska elettiin kesälomasesonkia, oli asiasta tietävien henkilöiden tavoittaminen haasteellista. Aikaa säästääkseen firma päätti tilata piirilevyn jysimisen Prinel Piirilevy Oy:ltä, jolla oli tarjota protokäyttöön soveltuvaa mallia firmalle sopivin kustannuksin sekä myös valmistaa se saman päivän aikana tilauksesta. Muutaman päivän kuluessa kortti olikin jo saapunut yrityksen tiloihin ja voitiin edetä puuttuvien komponenttien hankkimiseen, jotka olivat lähinnä riviliittimiä ja vastuksia.

Kun korttiin vaadittavat komponentit alkoivat olla selvillä, aloin laskea ja selvittää kunkin osan tarkkoja arvoja. Mikro-ohjaimen hallitsemaan osaan piiriä, johon kuului useita vastuksia sekä muutamia transistoreita, komponenttien arvojen selvittäminen oli varsin helppoa. Mikro-ohjaimen ohjeista selvisi, minkälaisia virta- ja jännitemääriä kortti kestää. Kun arvot olivat selvillä, oli varsin helppoa hakea läheisestä elektroniikkakaupasta korttiin soveltuvia vastuksia ja transistoreita. Moottorin käyttövirtaa ohjaava osa kortista oli hie- man monimutkaisempi, sillä kukaan ei osannut arvioida, paljonko virtaa sen kautta kulkisi rasituksen alla. Kortissa olevat releet tuli siis mitoittaa yläkanttiin. Tämä johti siihen, että releet tulivat verrattain kalliiksi ja toinen valitsemistani releistä ei ollut tarkoitettu piiri- korttiin juotettavaksi ainakaan sellaisenaan vaan sille olisi pitänyt olla jonkinlainen alusta. Oletan, että alustan tarkoitus olisi ollut mahdollistaa releen vaihtamisen tarpeen vaa- tiessa. Päädyin hoitamaan asian loppujen lopuksi siten, etten jyrsinyt piirilevyyn toiselle releelle paikkaa ollenkaan, vaan juotin kortista erilliset johtimet releelle. Lopputulos ei ollut visuaalisesti kaunis, mutta laite toimi. Laitteen ulkonäkö ei muutenkaan ollut mis- sään vaiheessa kynnyskysymyksenä, koska kyseessä oli prototyypin testaukseen suun- niteltu kortti, jonka kaikki komponentit päätyivät olemaan piilossa romupuusta tehdyssä laatikossa. Kun kaikki vaadittavat osat oli hankittu, pääsin kiinnittämään niitä piirikorttiin. Kiinnittämiseen käytin peruslaatuista käsikolvia (kuva 9), jonka vein yrityksen tiloihin kotoa, sillä yrityksellä itsellään ei ollut juottamiseen vaadittavia välineitä valmiina.



Kuva 9. Havainnekuva juottamisesta. Käyttäjä syöttää vasemmalla kädellään tinalankaa kuumennettuun komponenttien liitoskohtaan. Oikeassa kädessä kuumentamiseen käytettävä kolvi. (How to Solder Electronics 2017.)

Juottamisella tarkoitetaan metallisten pintojen liittämistä toisiinsa. Liittäminen tapahtuu sulattamalla liitoskohtaan tinaa, joka toimii siltana elektroneille ja pitää kappaleet yhdessä. Sulattaminen tapahtuu painamalla liitoskohtaa kuumennetulla juottimella (kolvilla) ja syöttämällä siihen haluttu määrä juotostinaa. Juottimen lämpötila riittää sulattamaan tinaa nopeasti, mutta ei vaikuta kuparista, hopeasta tai nikkelistä valmistettuihin liitoskohtiin. Kuitenkin jotkut, varsinkin pienemmät komponentit, voivat vaurioitua, mikäli ne altistuvat korkealle lämmölle pidemmäksi aikaa, joten juotoksen tulisi tapahtua mahdollisimman nopeasti. Kolvin kärki on hyvä tinata ennen juottamisen aloittamista, sillä se edesauttaa lämmön tasaista jakautumista juotosalueella. Käsikäyttöisen juottimen eli kolvin lisäksi voidaan tinaa sulattaa myös aaltojuottamalla, jota käytetään automatisoidussa piirilevyjen juotoksessa. (Juottaminen 2017.)

Juottamisessa on hyvä lisätä liitoskohtaan juoksutetta (fluksia) ennen tinaamista. Juoksuttimen tarkoitus on tasata kolvin tuomaa lämpöä tinattavalla alueella sekä poistaa liitoskohdasta juotokselle haitallisia aineita kuten metallioksidea. Tämä vahvistaa tinaa

ja komponentin välistä sidettä ja edistää näin lopputuloksen laatua. Juoksutin myös suo-
jaa liitoskohtaa myöhemmältä hapettumiselta. Työn valmistuttua on hyvä raaputtaa tai
pyyhkiä ylimääräinen kuivunut juoksutin kohteesta, sillä se voi aiheuttaa vuotovirtoja.
Kolvin kärkeä on myös hyvä putsata juottamisen aikana ja sen jälkeen, sillä siihen kertyy
jonkin verran epäpuhtauksia työn aikana. Kärkeä voi esimerkiksi pyyhkäistä välillä kos-
teaan pesusienien palaan. (Juottaminen 2017.)

Juottamisessa pitää huomioida muutamia riskikohtia. Terveydellisiin riskeihin kuuluu pa-
lovammojen vaara, kun käsitellään kuumaksi lämmitettyä kolvia sekä tinan sulattami-
sesta muodostuvat juotoshöyryt. Juottamisessa on siis huolehdittava ympäristöstään si-
ten, ettei lähettyvillä ole helposti syttyvää materiaalia ja että alue on hyvin ilmastoitu.
Tämän lisäksi jotkut tinat, pääasiassa lyijypitoiset, voivat olla vaaraksi paljaalle iholle ja
niiden koskettamista tulisi välttää. Myös itse juottamisessa tulee olla tarkka, sillä vaikka
tina sulaisi ja päällisin puolin näyttäisi yhdistävän tinattavaa metalliliitosta, ei se välttä-
mättä tarkoita, että liitos on onnistunut. Sähköä johtavien pintojen ja tinan väliin on
voinut jäädä epäpuhtauksia, jotka voivat vaikuttaa haitallisesti virrankulkuun joko lyhy-
ellä tai pitkällä aikavälillä. (Juottaminen 2017.)

Tein muutamia huolimattomuusvirheitä juottamisen aikana. Olin transistoreita juottaessa
katsonut väärän transistorityypin ohjekirjasta, ja se johti sekaannukseen transistorin jal-
kojen järjestyksestä. Huomasin tämän, kun väärin kytketyt transistorit eivät testauksessa
toimineet kuin oletin ja ilmeisesti hajosivat, koska kun juotin jalat oikeille paikoilleen, niin
toiminta jatkui virheellisenä. Päädyin vaihtamaan transistorit uusiin. Onneksi transistorit
ovat halpoja ja niitä oli hankittu ylimääräisiä. Hieman samankaltaisen virheen tein vir-
ranmittauspiirin kanssa, jossa sekoitin pari jalkaa toisiinsa. Olin katsonut kyllä oikeaa
ohjekirjasta, mutta kaavakuvaan, josta jalkojen paikka selviää ei ollut selkeästi merkitty,
miten päin komponentti oli kuvassa esitetty. Opin, että komponenttien kanssa saa olla
tarkkana, sillä katsomani transistorin nimi erosi oikean transistorin nimestä yhdellä kir-
jaimella ja virta-anturin ohjeet olivat ehkä totuttua epäselvempiä.

2.3 Datankäsittely

Jotta releitä ohjaavat transistorit osaisivat ajaa sähkömoottoria oikean ajan oikeaan suuntaan, ne tarvitsivat ohjainkomponentin. Ohjainkomponentiksi valittiin mikro-ohjain, jonka tarkoituksena olisi myös kerätä ja lähettää kortin virtamittarin dataa eteenpäin tietokoneelle. Mikro-ohjaimella käsiteltäisiin myös laitteeseen asennettavien kytkinten signaaleja ja niiden perusteella ohjaimen prosessori ohjaisi releiden yhteyteen asennettuja transistoreita johtaviksi. Tietokoneella datankäsittelyyn vaaditaan lukuohjelma, joka osaa kääntää mikro-ohjaimelta tulevan datan ymmärrettäväksi tekstiksi sekä kykenee tallentamaan sen tietokoneelle tekstitiedostoon.

2.3.1 Mikro-ohjaimet

Mikro-ohjaimet tai mikrokontrollerit ovat käytössä lähes kaikissa monimutkaisemmissa elektronisissa laitteissa. Mikro-ohjain on käytännössä pieni tietokone, jossa on yksi tai useampi prosessori eli suoritinyksikkö, muistia sekä ohjelmoitavia sisään- ja ulostulo liitäntöjä. Lähes kaikissa mikro-ohjaimissa on useammanlaista muistia. Ohjaimen kiinteä muisti, joka sisältää suoritettavan ohjelman, sekä väliaikainen muisti jonka avulla ohjain tallettaa ja käyttää tietoja. Mikro-ohjainten pääasiallinen käyttöalue on automatisoiduissa laitteissa kuten työkaluissa ja -laitteissa, joihin ne soveltuvat hyvin pienen kokonsa sekä pienen virrankulutuksensa takia. (Microcontroller 2017.)

Mikro-ohjaimet hyödyntävät niin digitaalisia kuin analogisiakin signaaleja käyttötarkoituksestaan riippuen. Digitaalisen signaalin arvo on joko yksi tai nolla, kun taas analogisen signaalin arvo vaihtuu portaattomasti. Vaikka mikro-ohjaimet eivät viekään yleensä paljon käyttövirtaa, vaihtelee sekin käyttötarkoituksen mukaan. Matalia taajuuksia hyödyntävä heikkotehoinen mikro-ohjain sopii hyvin patterilla tai akulla toimiviin laitteisiin, kun taas voimakkaalla prosessorilla ja korkealla kellotaajuudella varustettu monimutkaisempia toimintoja suorittava ohjain vie enemmän virtaa. (Microcontroller 2017.)

Mikro-ohjaimia sovelletaan upotettuihin järjestelmiin, eli järjestelmiin joilla on täsmällinen tehtävä osana suurempaa mekaanista tai elektronista systeemiä. Upotetut järjestelmät hyötyvät ohjainten pienestä koosta, pienestä virrankulutuksesta sekä monipuolisuudesta. Mikro-ohjain säättää yleensä laitteessa olevia kytkimiä, releitä, solenoideja ja voi

käsitellä antureita, radiotaajuussovelluksia sekä näyttöpäätteitä. Sulautetuissa järjestelmissä on harvoin minkäänlaista käyttäjälle tarkoitettua päätettä, mistä laitteen toimintaa voisi muokata. Mikro-ohjaimen käyttämä koodi asennetaan laitteeseen siis ja rakennusvaiheessa. (Microcontroller 2017.)

Mikro-ohjaimessa on usein kymmenittäin erinäisiä liitäntöjä, jotka voidaan määritellä sisään- tai ulostuloiksi. Näiden liitinten kautta kulkevat signaalit voivat olla joko analogiasia tai digitaalisia ohjelmoinnin ja tarpeen mukaan. Analogisia signaaleja, kuten anturiarvoja, lukiessaan sulautetussa järjestelmässä on oltava osa, joka mahdollistaa analogisen signaalin muunnoksen digitaalseksi (A/D-muunnos). Prosessori ei itsessään pysty käsittelemään kuin digitaalisia signaaleita. On olemassa myös D/A-muunnokseen pystyviä osia joilla prosessori voi lähettää analogista signaalia. Suurimmassa osassa mikro-ohjaimia on myös yksi tai useampi ajastin, jota voidaan käyttää laitteen toiminnan tahdittamiseen tai muuhun ajasta riippuvaan toimintaa kuten lämpötilan mittaamiseen tietyin aikavälein. (Microcontroller 2017.)

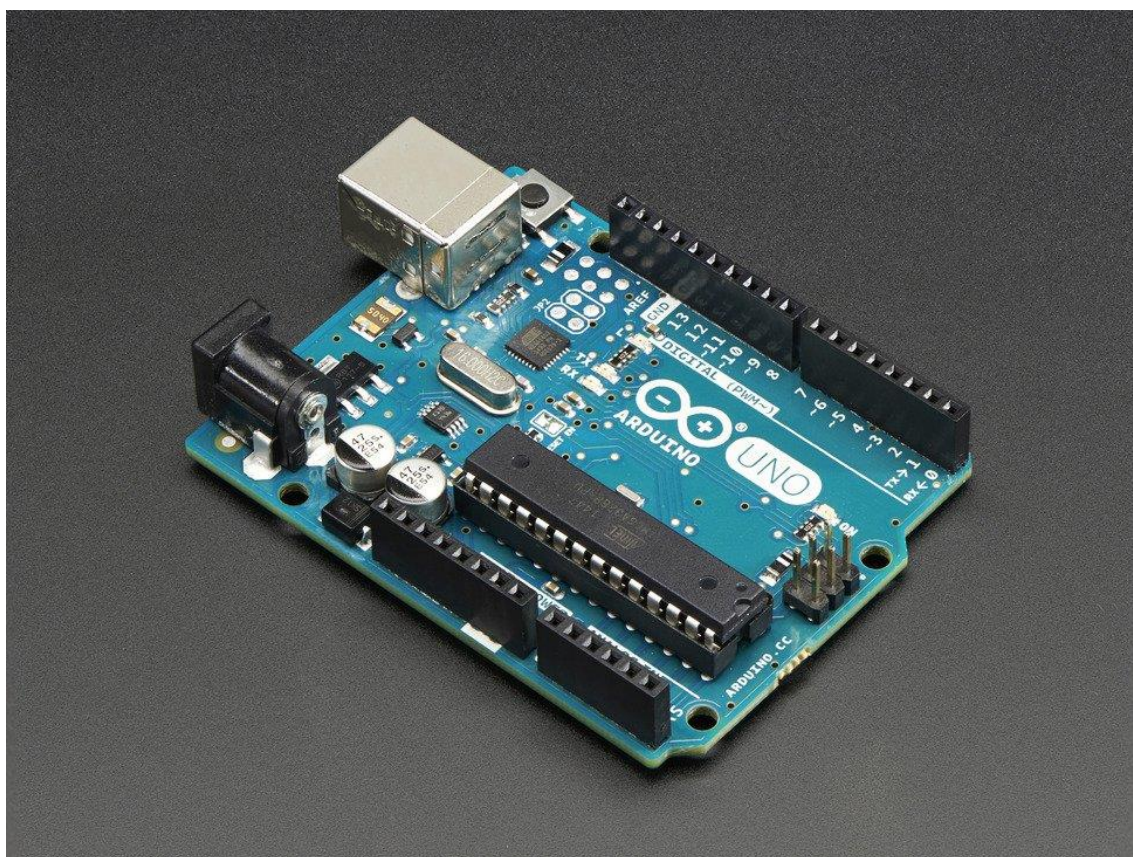
Mikro-ohjaimia valmistetaan ja käytetään pääsääntöisesti siten, että ohjaimeen on integroitu ylimääräistä teknologiaa kuten muistia tai liitäntöjä. Vaikka ylimääräisillä osilla valmistettu ohjain tulee valmistusvaiheessa kalliimmaksi kuin ohjain, joka käyttäisin tarkan määrän kortin ulkopuolella olevia resursseja, tulee se pitkällä aikavälillä halvemmaksi. Hyöty muodostuu harvempien ohjaimien käyttämisestä joka pienentää piirikortin kokoa, ohjaimen yksinkertaistetusta testaus- ja rakentamisvaiheesta sekä vähentyneistä virheellisten ohjainten määrästä. Kyse on siis jossain määrin teknologian standardoinnista eli yhtenäistämisestä. (Microcontroller 2017.)

2.3.2 Arduino

Arduino on omaa avointa lähdekoodia hyödyntävä ohjelmoitava mikro-ohjain, joka on suunniteltu helppokäyttöiseksi ja sopii hyvin prototyyppi- ja testilaitteisiin. Arduinon ohjelmoinnissa käytetään sen omaa Arduino programming language -kieltä, joka pohjautuu toiseen mikro-ohjaimille tarkoitettuun ohjelmointikieleen, Wiringiin. Arduino tuoteperheeseen kuuluu toistakymmentä erilaista mikro-ohjainkorttia, joista päädyimme käyttämään Arduino Unoa sen helppokäyttöisyyden ja huokeyn hinnan takia. Lisäksi Arduinoa oli käytetty myös joillakin koulun ohjelmointiin ja mittauksiin liittyvillä kursseilla, joten

minulla oli jonkin verran kokemusta sen käytöstä ja tiesin sen kykenevän tarvittaviin tehtäviin suhteellisen helposti. (What is Arduino 2017.)

Arduino Uno (kuva 10) on itsenäinen mikro-ohjainkortti, jossa on 14 digitaalista sisään- ja ulostuloporttia ja 6 analogista sisääntuloporttia. Unoa ei ole tarkoitus juottaa kiinni piirikorttiin vaan ulos- ja sisääntuloina toimivat liittimet. Tämä nopeuttaa kortin asentamisen lisäksi myös kortin vaihtamista, mikäli kortti vaurioituu. Uno saa käyttöjännitteensä joko USB-johdon kautta tietokoneesta tai muuntajan välityksellä pistokkeesta. Testilaitteessa Arduino Uno oli kytketty kannettavaan tietokoneeseen USB-johdolla, jota pitkin myös siirrettiin virta-anturin keräämää dataa.



Kuva 10. Arduino Uno mikro-ohjain. (Arduino Uno R3 (Atmega328 - assembled) 2017.)

Arduino on avoimen lähdekoodinsa ansiosta helppo opetella, sillä valmiita ja puolivalmiita esimerkkitoiteutuksia koodeineen on tarjolla lukuisia. Yksinkertainen ohjelmointikieli (kuva 11) on esillä Arduinon kotisivuilla ja eri kommentojen yhteyteen on lisätty lyhyitä esimerkkejä selityksineen. Sivun foorumeilta löytyy myös lukuisia käyttäjien esimerkkejä

ja harrastajat jakavat usein pätkiä koodistaan esimerkiksi Githubissa tai vastaavissa ohjelmoijien keskustelupalstoilla.

```
int RELE1 = 6;
int RELE2 = 7;
int KYTKINVIRTA = 5;
int KYTKIN1 = 2;
int KYTKIN2 = 3;
int KYTKIN3 = 4;
float Virtadata = A0;
float VirhemA = 3220;

void setup() {

    Serial.begin(9600);
    pinMode (RELE1, OUTPUT);
    pinMode (RELE2, OUTPUT);
    pinMode (KYTKIN1, INPUT);
    pinMode (KYTKIN2, INPUT);
    pinMode (KYTKIN3, INPUT);
    pinMode (KYTKINVIRTA, OUTPUT);
    pinMode(Virtadata, INPUT);
    char buffer[33];

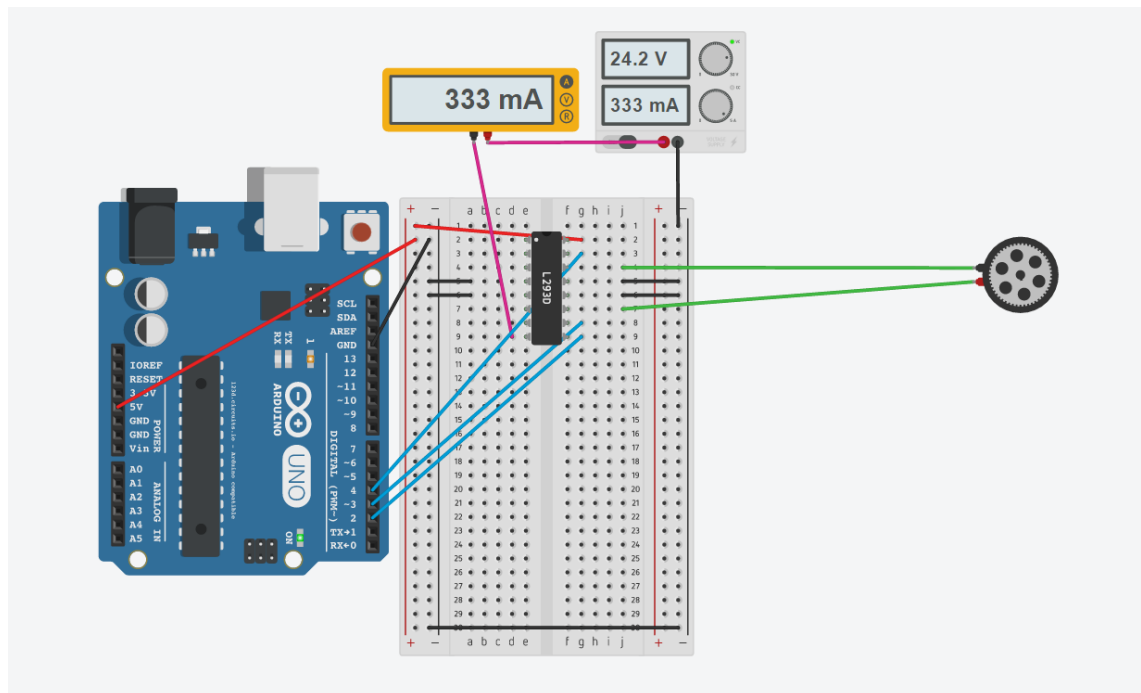
    digitalWrite(RELE1, LOW);
    digitalWrite(RELE2, LOW);
    digitalWrite(KYTKIN1, LOW);
    digitalWrite(KYTKIN2, LOW);
    digitalWrite(KYTKIN3, LOW);
    digitalWrite(KYTKINVIRTA, HIGH);

}
```

Kuva 11. Arduinon sisältämän moottorinohjauskoodin alussa määritellyjä komentoja.

Kun Arduino oli valittu, lähdin etsimään internetistä jonkinlaista työkalua, jolla voisi harjoitella niin Arduinon ohjelmoimista kuin hahmotella piirilevyn kokonaisuutta. Löysin nopeasti ohjelman nimeltä Autodesk Circuits (kuva 12), jossa oli juurikin etsimäni ominaisuudet. Autodesk Circuitsissa oli mahdollista käyttää useiden Arduino-ohjainten lisäksi lähes kaikkia elektroniikassa käytettäviä komponentteja, sekä liittää ne toisiinsa virtuaalisin johtimin virtuaalisella koekytkentälevyllä. Kaiken lisäksi Arduinolle pääsi ohjelmassa kirjoittamaan koodia, jolla onnistuin pienen harjoittelun jälkeen ajamaan virtuaalista kaksisuuntaista tasavirtamoottoria 5 sekuntia yhteen suuntaan ja vaihtamaan suuntaa sen jälkeen. Yksinkertaistettu simulaatio sisälsi vain Arduinon, moottorin, suunnanvaihtamiseen tarkoitetun H-sillan sekä ulkoisen virtalähteen ja virtamittarin. Tärkeintä oli saada

kuva laitteen toiminnasta ja eräänlainen todiste sen toimivuudesta. Simulaation jälkeen oli helppoa lähteä hahmottelemaan vaadittavia yksityiskohtia niin itsenäisesti kuin ohjaavan opettajan ja FLX-Framen kanssa. Autodesk Circuitsissa olisi myös mahdollista piirtää ja tilata valmiita piirilevyjä, mutta koska minulla oli mahdollista koulun kautta suunnitella se PADS:illa, jätin mahdollisuuden käyttämättä.



Kuva 12. Autodesk circuitsissa simuloitua sähkömoottorin suunnanvaihtamista H-sillan avulla.

Suunnittelun aikana piirsin myös useasti paperille ajatuksia niin piirikortista kuin Arduinon ohjelmoinnista. Varsinkin ohjelmointia selventäviä laatikkokaavioita tein useita ja niiden tarkoituksena oli yksinkertaisesti näyttää minulle ja muille Arduinon ohjelman toimintajärjestystä. Kaavioista näkyi esimerkiksi, miten Arduino reagoi, kun rajakytkimen arvo muuttuu. Tekemäni kaaviot eivät juurikaan noudattaneet tarkkaa mallia, joita on käsittääkseni olemassa useita, vaan olivat enemmänkin muistiinpanoihin verrattavia tuherruksia. Virallisista malleista olisi varmasti apua monimutkaisempia koodeja kirjoittaessa, mutta Arduinossa käytettävä ohjelma oli varsin yksinkertainen eikä käsittääkseni kukaan firman puolelta tuntenut ohjelmointia niin hyvin, että virallisemmasta mallista olisi ollut mainittavaa hyötyä.

2.3.3 Virran mittaus

Laitteen testauksessa oli tarkoitus mitata virrankulutusta ja alusta asti oli suhteellisen selvää, että mittaus toteutettaisiin siihen tarkoitettuun integroidun piirin avulla. Integroidulla piirillä tarkoitetaan piiriä, joka on pakattu yhdeksi komponentiksi. Integroidun piirin sijasta mittaamisen olisi varmasti voinut hoitaa myös mittaamiseen tarkoitettulla laitteella kuten lämpötilan seuranta päätettiin toteuttaa, mutta mittarin, joka voisi myös tallentaa mittaamansa tulokset hankkiminen olisi tullut merkittävästi kalliimmaksi. Integroidun virtamittarin ohjekirjasessa oli hyvät vaikkakin hieman sekavat ohjeet komponentin käyttöön. Kytkentäkaavion lisäksi ohjeista löytyi kaavio, jonka avulla komponentin Arduinolle lähettämän jännitesignaalin saa muutettua ampeereiksi. Lisäksi ohjekirjassa oli suosituksia integroidun piirin yhteyteen asennettavista komponenteista, joilla mittaus-tarkkuutta saadaan entisestään parannettua.

2.3.4 Tiedon keräys ja tallentaminen

Datan siirtäminen Arduinolta tietokoneelle ja toisinpäin on varsin yksinkertaista. Data on kuitenkin siirrettäessä binäärimuodossa, ja se tarvitsee kääntämisohjelman, jotta se saadaan muutettua tekstimuotoon. Arduinoa ohjelmoidessa ohjelmointiympäristö muuntaa datan siirtoa varten mikro-ohjaimelle sopivaan muotoon, mutta kun kortilta lähetetään esimerkiksi anturidataa takaisin, se vaatii toisen ohjelman vastaanotossa. Pyrin aluksi hoitamaan datan vastaanoton ja tiedostoon tallentamisen Arduinolla, mutta asiaa tutkituani päädyin siihen tulokseen, että se olisi minun ohjelmointitaidoillani lähes mahdotonta. Huomasin kuitenkin Arduinon keskustelupalstoilla jonkun ehdottaneen Gobetwinonimisen ohjelman käyttöä. Ohjelma kykenisi teoriassa vaatimiini toimenpiteisiin, mutta käyttäjä keskustelupalstalla varoitti sen myös olevan hieman vanhentunut ja keskeneräinen. Päätin kuitenkin kokeilla ohjelmaa, sillä se näytti yksinkertaiselta käyttää, eikä minulle tullut muitakaan ajatuksia ongelman ratkaisemiseksi.

Gobetwino on tietokoneelle ladattava ohjelma, joka on luotu yksinkertaistamaan datan muuntamista ja tiedostoon tallentamista käyttäen hyväksi ohjelmaan rakennettua kirjastoja. Testilaitteessa Gobetwino luki Arduinon keräämän datan virtasensorista ja tallensi sen tekstitiedostoon myöhempää analysointia varten. Samalla tallennettiin myös mittauksen aika, jonka Gobetwino luki kannettavan kellosta. Gobetwinolle voisi myös antaa

käskyn lähettää tuloksia suoraan sähköpostitse, mikäli tietyt kriteerit täyttyvät, tai sille voi antaa ohjeita sähköpostitse. (Gobetwino 2017.)

Gobetwino toimii käytännössä siten, että Arduinon sensorimittauskoodin yhteyteen lisätään Gobetwinon määrittämä koodi (kuva 13). Koodissa on ohjeet Gobetwinolle mitä mitatulla datalla tulee tehdä, esimerkiksi tallennetaanko se tiedostoon tai lähetetäänkö sähköpostina jollekin. Testilaitteessa tallensin virranmittauksen ja mittausajankohdan Windowsin mukana olevaan Wordpad-tekstinkäsittelyohjelmaan, josta tein myös pikakuvakkeen koneen työpöydälle, jotta yrityksen jäsenet voisivat kätevästi itse käydä lukemassa mitattuja arvoja, mikäli tulisi tarve.

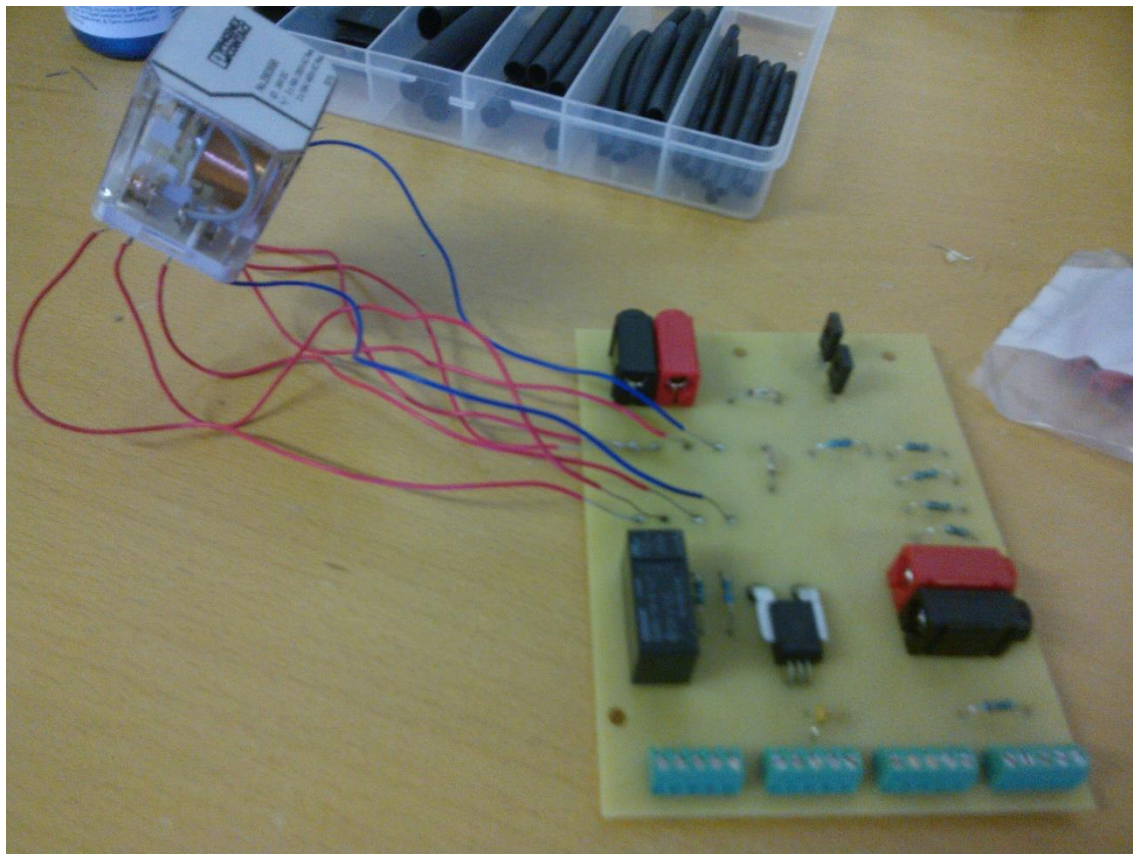
```
a = 1;
digitalWrite(RELE1, HIGH);
if (b < a)
{
  delay (3000);
  float Virta = (((analogRead(Virtadata) * (5.0 / 1023.0)) / 0.040) * 1000 + VirhemA)); //Testattu 0A antaa luvuksi 3A n. 130mV ulostulo
  char buffer[33];
  Serial.print("#S|VIRTA|ESTI|");
  Serial.print(itoa((Virta), buffer, 10));
  Serial.print(" mA");
  Serial.println("]#");
  b = 1;
}
```

Kuva 13. Arduinon Gobetwinoa hyödyntävä koodinpätkä.

2.3.5 Moottorin pyörimissuunnan vaihtaminen

Arduinolle tulevien kolmen kytkimen signaaleita käytettiin moottorin ohjaamiseen. Laitteessa oli asennettuna yksi ON/OFF-kytkin, sekä kaksi rajakytkintä. ON/OFF-kytkintä käytettiin nimensä mukaisesta aloittamaan prototyyppilaitteen ajo. Rajakytkimet oli asetettu ajettavan laitteen ylä- ja ala-asennon kohdille siten, että laitteen ajon yhteydessä ajettava massa työnsi kytkimen ON-asentoon. Arduino havaitessaan jommankumman kytkimen signaalin, syöttää releiden yhteyteen asennetuille transistoreille jännitteen, joka sai releen muuttamaan asentoaan. ON/OFF-kytkimen rele oli SPDT-tyyppiä, jonka toinen ulostulo ei ollut kytkettynä mihinkään. Arduinon havaitessa ON-asennon releen kautta kulki virtaa piirissä olevalle suunnanvaihtoreleelle ja sen kautta moottorille. Suunnanvaihtorele DPDT (kuva 14) oli yhdistetty moottorin virtalähteenä toimivaan vapaasti säädettävään tasajännitelähteeseen ja sen maahan. Releen neljä ulostuloa oli kytketty ristiin siten, että releen aktivoituessa virran kiertosuunta muuttui, joka sai vuorostaan

sähkömoottorin pyörimään eri suuntiin. Suunnanvaihtoreleen asento määräytyi rajakyt-
kinten signaaleista prosessorille, siten että ajon alussa moottoria ajettiin yhteen suun-
taan ja kytkimen kytkeytyessä suuntaa vaihdettiin.



Kuva 14. Piirikortti releineen. Vasemmalla alhaalla kortissa SPST rele ja kytkinlankojen varassa oleva SPDT rele.

Arduinon ohjelmassa sunnanvaihtosignaalien käsittelyn yhteyteen koodattiin myös pieni delay -komento, jonka jälkeen Arduino luki virtamittauksen piirissä olevalta virta-antu-
rilta. Delayn tarkoitus oli eliminoida mahdollisia virtapiikkejä heti sunnanvaihdon jälkeen
ja näin tuottaa tasaisempaa dataa virrankäytöstä. Virranmittaus ja sen tulosten tallenta-
minen tiedostoon oli yksi minulle yksi hankalampia osia työssä, sillä se sisältää monimut-
kaisempaa koodia kuin pelkkä transistoreiden ohjausvirran ja kytkinten logiikan koodaus
oli vaatinut. Niin ikään virranmittaus oli myös yksi tärkeämpiä osia laitteessa, sillä kerät-
tyä tietoa voidaan käyttää myöhempiä protoja ajatellen. Yhden ajon aikana kerättiin
kaksi virran mittaustulosta. Käytännössä testilaitte käytti sähkömoottoria jousien työntä-
miseen ja palauttamiseen, joten se kulutti merkittävästi enemmän virtaa työskennelles-
sään jousia vastaan kuin palautuksessa, jossa jouset tekivät suurimman osan työstä.

Vaikka jousien ja niiden päässä olevan kuorman siirto oli raskaampaa ja näin ollen hyödyllisempää dataa, päätettiin uteliaisuudesta seurata myös palautuksessa käytettävää virtaa.

3 Testaus

Testauksen (kuva 15) aikana esiintyi hankaluuksia sähkömoottorin liittimen kanssa ja lämpöanturin lisäämisessä piiriin. Käyttämämme sähkömoottorin mukana ei tullut soveltuvia liittimiä. Kauppoja ja Internetiä kiertelemälläkään ei löytynyt sopivan näköisiä ja yksi asiaa selvittävistä taisi sanoa nähneensä hieman samankaltaisen liittimen Itä-Saksan aikaisesta dokumentista. Liitin päätettiin lopuksi rakentaa itse ottamalla mahdollisimman tarkat mitat moottorin liittimestä työntömitalla ja rakentamalla peruslaatuisille puristinliittimille kotelo. Piirikortin valmistumisen jälkeen keksittiin, että lämpötilamittarin asentamisesta suunnanohjauspiiriin olisi ollut hyötyä. Aluksi pohdittiin halvan lämpötilamittarin asentamista piirikorttiin ja Arduinoon jälkikäteen, mutta lopulta päädyttiin käyttämään yritykseltä löytyvää vanhaa anturia. Anturi oli suunniteltu lämmön mittaamisen lisäksi myös tietojen kirjaamiseen tietokoneelle. Valitettavasti Anturin mukana tullut softa ei sellaisenaan toiminut nykyisellä käyttöjärjestelmällämme ja toimivan hankkimista jouduttiin selvittämään useista paikoista. Soittamalla paikalliselle jälleenmyyjälle selvisi thaimaalainen valmistaja ja ottamalla heihin yhteyttä saatiin hankittua Windows 7:llä toimiva versio anturin ajurista. Kyseistä anturia ei ollut myyty ainakaan Suomessa päälle kymmeneen vuoteen, mikä selitti ongelmamme.



Kuva 15. Testauksessa hajonnut virtalähde.

Testauksen aikana esiintyi muitakin ongelmia niin laitteen rakentamisessa, asennuksessa ja ajamisen aikana. Ongelmat vaihtelivat kokemattomuudesta johtuvista virheistä erinäisiin laitevikoihin. Suurin osa virheistä pystyttiin ratkaisemaan tekemällä tarkoitettu asia toisin kuin oli ensin ajateltu ilman, että ongelma olisi aiheuttanut merkittävää viivästystä työn etenemiseen. Suunnittelun osalta virheitä tuli PADS:issa toisen releen jalkojen paikkojen mittauksessa ja ohjelmaan syöttämisessä, jonka takia ne eivät osuneet aivan kohdalleen. Jalat sai kuitenkin taitettua sen verran sivuun, että juottaminen onnistui. Toinen PADS:issa tapahtunut virhe oli merkittävästi vakavampi. Koska tiedossa ei ollut, millaisia virtamääriä kortin kautta johdettaisiin moottorille, jätin kortin johdinpintojen leveyden lähelle oletusasetuksia. Oletusleveydet toimivat hyvin pienillä virtamäärillä, mutta moottorin pakkasajotestissä, jossa moottorin pumpun hydraulikkaneeste jäätynyt kokonaan ja moottori yritti pumpata nestettä tästä huolimatta käyttäen suurta virtaa, kortin johdin paloi poikki. Jälkikäteen mietittynä olisi tietenkin pitänyt asettaa johdinleveydet kestävämmän paljon suurempia virtoja varmuuden vuoksi. Kuitenkin testin perimmäisenä tarkoituksena oli nähdä, mitkä osat pettävät ääriolosuhteissa ja sen kannalta molemmat: joh-

timen poikki palaminen ja hydraulikkaneesteen jäätyminen olivat arvokasta tietoa tulevien prototyyppien kehityksessä. Testausta voitiin tämänkin jälkeen jatkaa huoneenlämmössä, kun korvasin piirikorttiin jyrityn johtimen kytkentälangalla.

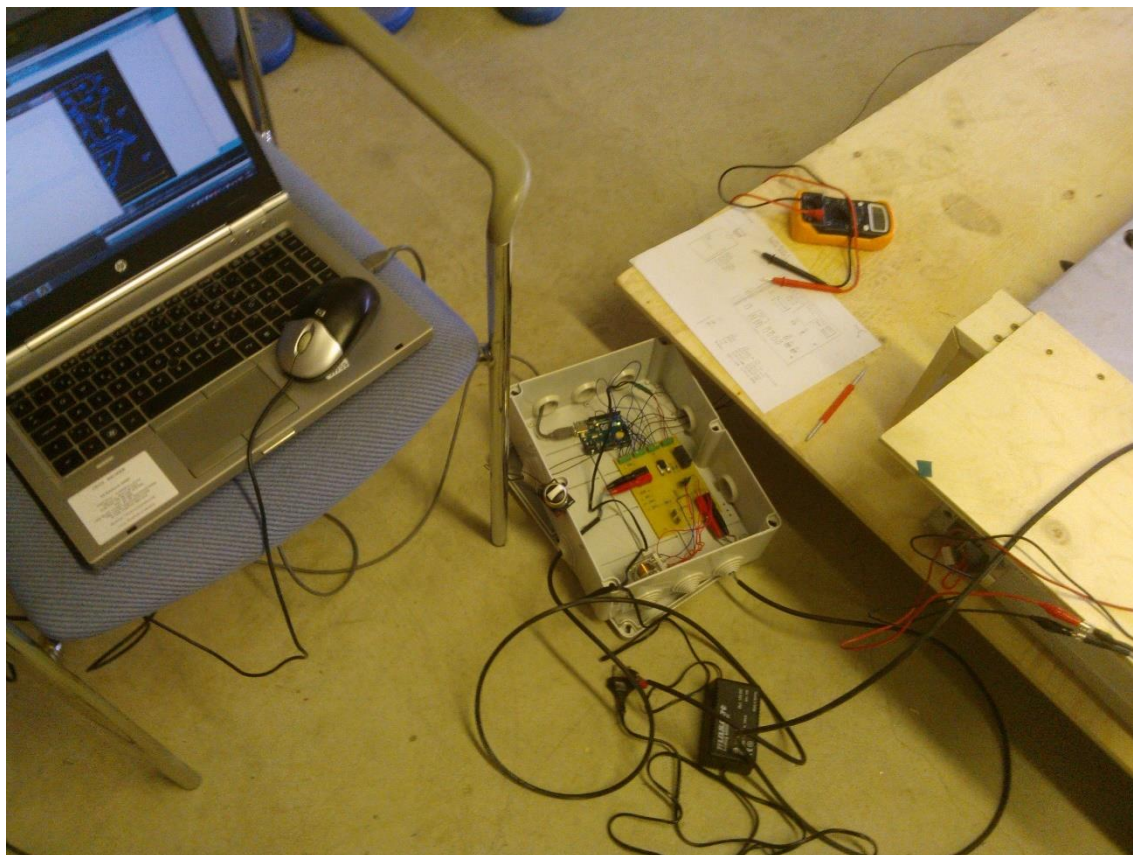
Koodatessani Arduinoa käytin delay-komentoa eli komentoa, joka saa ohjelman odottamaan määrätyn ajan ennen toiminnan jatkamista. Delay sijoitettiin ohjelmassa virranmittauksen yhteyteen, jotta saatiin tasainen lukema moottorin käyttämästä virrasta. Huomasin komentoa käyttäessäni, että delay vaikutti muuhun koodin toimintaa sikäli haitallisesti, että Arduino ei sen aikana lukenut esimerkiksi rajakytkimien kautta meneviä signaaleja, joiden avulla ohjattiin moottorin suuntaa. Vaarana siis oli, että jos delaytä olisi liikaa tai se olisi sijoitettu huonoon kohtaan koodia, voisi se saattaa aikaan virhetilanteen. Hoidin asian rajoittamalla delayn keston mahdollisimman pieneksi ja tarkistamalla, ettei sen aikana olisi meneillään mitään prototyyppilaitteen liikkeeseen vaikuttavaa komentoa.

Gobetwinonkaan käyttö ei suinkaan ollut ongelmaton, sillä keskeneräistä yhden henkilön tekemää ohjelmaa ei ole päivitetty vuoden 2011 jälkeen. Ohjelma oli herkkä kaatui- lemaan, mikäli toimintoa tai asetuksia yritettiin muokata ja vaati kehitysvaiheessa uudelleenkäynnistämistä jatkuvasti. Itse testauksen aikana laite kuitenkin toimi hyvin ja kirjasi mitatut arvot onnistuneesti läpi yön kestävässä mittauksissa. Viikonlopun yli kestävässä mittauksissa oli ajoittain jotakin ongelmia, kuten muutamien mittaustulosten häviäminen. Selkeää syytä tähän ei löytynyt, mutta itse laitteen ajo ei koskaan keskeytynyt, joten voidaan olettaa, että Gobetwino oli hetkellisesti jumiutunut. Muutamien mittaustulosten häviäminen useista sadoista ei kuitenkaan ollut äärimmäisen vakavaa, sillä jäljelle jääneillä tiedoilla saatiin tarpeeksi tarkkaa kuvaa virrankulutuksesta.

4 Yhteenveto

Kokonaisuudessa (kuva 16) lopputyössä suunniteltiin ja valmistettiin kaksisuuntaisen sähkömoottorin ajamiseen soveltuva piirikortti, jossa virran suunnanmuutos oli hoidettu mikro-ohjainta, transistoreita ja releitä hyödyntämällä. Kortissa oli myös useita suurempia ja pienempiä liittimiä, joista pienempiä käytettiin Arduinon toimintojen kanssa ja suurempia moottorin ja päävirtalähteen sisään- ja ulostuloreitteinä. Korttiin oli lisätty myös useita vastuksia, diodeita ja kondensaattori tukemaan erinäisiä elektronisia toimintoja.

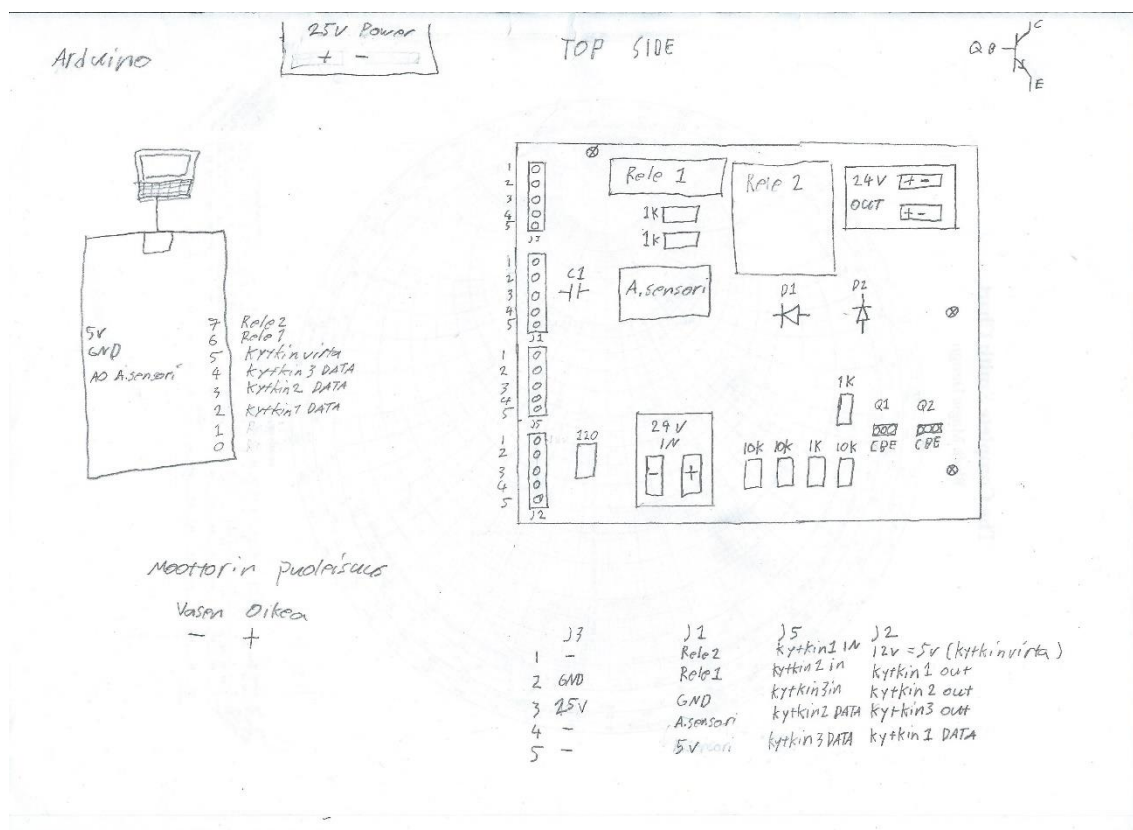
Piirikortti on liitetty Arduino Uno -mikro-ohjaimeen, jolla hoidettiin releiden asennonvaihtoa kytkinsignaalien avulla, sekä otettiin mittaustuloksia käytetystä virtamäärästä. Mittaustulokset tallennettiin Arduinoon liitettyyn kannettavaan tietokoneeseen, jonka päässä Gobetwino -ohjelma kirjoitti datan tekstitiedostoon myöhempää käsittelyä varten merkiten samalla muistiin kirjaamisajankohdan.



Kuva 16. Lopputyön rakennusvaiheessa oleva kokonaisuus. Vasemmalla ohjelmointiin ja tiedonkäsittelyyn käytetty tietokone, keskellä moottorinohjainkortti ja oikealla itse testilaite.

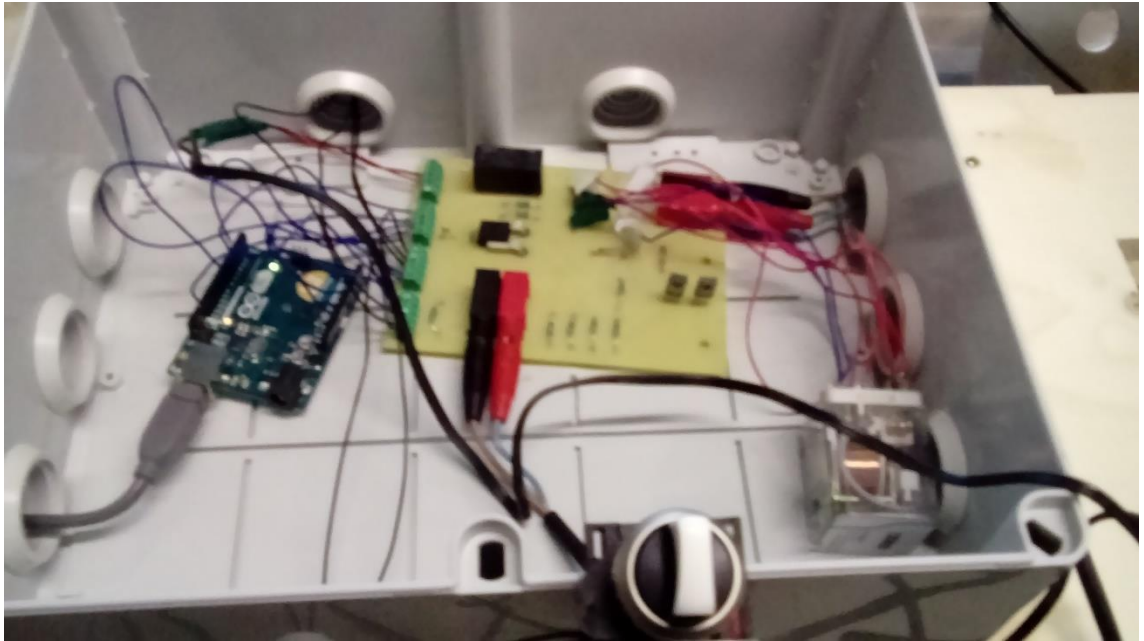
Työn aikana saavutettiin projektin alussa asetetut tavoitteet. Valmistettiin laite (kuva 18), joka vaatimusten mukaisesti ohjasi hydraulikkapumppua pyörittävää kaksisuuntaista sähkömoottoria ja keräsi tietoa virrankulutuksesta ja lämpötila-arvoista sekä tallensi ne laitteeseen kytkettyyn tietokoneeseen. Kerättyä tietoa on käytetty ja tullaan käyttämään tuotteen jatkokehityksessä ja tulevilla testeillä. Testejä suoritettiin pääasiassa FLX-Framen tiloissa, mutta myös ulkopuolisen tahon pakastehallissa. Testien aikana törmättiin erinäisiin ongelmiin ja haasteisiin, joita ratkottiin työn aikana tai tullaan ratkaisemaan myöhemmissä prototyyppiversioissa. Testaamisen osalta työ jatkuu lopputyön suorittamisen jälkeen osaksi käyttäen hyväksi rakentamaani korttia ja sillä kerättyä

tietoa. Yritykselle on jätetty myös useita ohjainkortin toimintaa selittäviä piirustuksia (kuva 17) sekä käyttöohjeet, jotka edesauttavat laitteen käytössä.



Kuva 17. Piirtämäni kuva kortista kytkentäohjeineen.

Opin työn aikana merkittävästi piirikorttisuunnittelusta ja myös tiedon soveltamisesta ongelmien sattuessa kohdalle. Joitakin asioita oli opeteltu koulussa, kuten PADS:in käyttöä ja ohjelmoinnin perusteita. Näillä eväillä ei työssä päässyt kuitenkaan mahdollottoman pitkälle, mutta ne osoittivat mistä, voidaan lähteä liikkeelle suunnitteluprosessissa, joka on ollut itselleni monesti se hankalin vaihe. Kun suunnittelu oli lähtenyt liikkeelle, oli paljon helpompi lähteä miettimään yksityiskohtia ja käyttämään hyväkseen tarjolla olutta tietoa. Osa asioista taas oli lähes kokonaan uutta, kuten kerätyn tiedon tallentaminen tietokoneelle Arduinoa käyttämällä ja releiden toiminta. Näiden lisäksi pääsin tutustumaan hieman mekaanisiin ja hydraulisiin järjestelmiin ja näkemään, miten ne toimivat elektronisten järjestelmien kanssa.



Kuva 18. Viimeistelyä vaille toiminnassa oleva ohjainkorttipaketti.

Parannettavaa työstä olisi ehdottomasti huolellisuus suunnitteluvaiheessa, sillä ne muutamat pieniltä tuntuvat virheet aiheuttivat huomattavia ongelmia pitkällä aikavälillä. Tiedostin tämän jo projektin aikana, mutta en jälkikäteen mietittynä osannut suhtautua siihen tarpeeksi vakavasti. Myös jatkuvaa vuoropuhelua olisin voinut harrastaa enemmänkin. Esimerkkinä jälkikäteen sovellettu lämpötilanmittaus, joka olisi suunnitteluvaiheessa ollut suhteellisen helppo sisällyttää piirikortille ja hoitaa datankäsittely Arduinolla.

Lähteet

Arduino Uno R3 (Atmega328 - assembled). Verkkodokumentti. Adafruit.

<<https://www.adafruit.com/product/50>>. Luettu 6.5.2017.

Automatic Staircase Lights using PIR Sensor and Relay. 2017. Verkkodokumentti. CircuitDigest.

<<https://circuitdigest.com/electronic-circuits/automatic-room-lights-using-pir-sensor-and-relay>>. Luettu 6.5.2017.

FR-4. 2017. Verkkodokumentti. Wikipedia. <<https://en.wikipedia.org/wiki/FR-4>>. Päivitetty 23.2.2017. Luettu 3.4.2017.

Gobetwino. Verkkodokumentti. Mikmo. <<http://mikmo.dk/gobetwino.html>>. Luettu 1.4.2017.

GoBetWino 0.6 on the way – for real. 2011. Verkkodokumentti. mikmo.

<<http://mikmo.dk/blog/>>. Luettu 7.4.2017.

How to Solder Electronics. 2017. Verkkodokumentti. wikiHow.

<<http://www.wikihow.com/Solder-Electronics>>. Luettu 6.5.2017.

Juottaminen. 2017. Verkkodokumentti. Wikipedia.

<<https://fi.wikipedia.org/wiki/Juottaminen>>. Päivitetty 20.2.2017. Luettu 3.4.2017.

Microcontroller. 2017. Verkkodokumentti. Wikipedia. <<https://en.wikipedia.org/wiki/Microcontroller>>.

Päivitetty 8.2.2017. Luettu 3.4.2017.

PADS Standard. 2015. Verkkodokumentti. Mentor Graphics.

<http://s3.mentor.com/public_documents/datasheet/pads_com/standard-ds.pdf>. Luettu 1.4.2017.

Rele. 2016. Verkkodokumentti. Wikipedia. <<https://fi.wikipedia.org/wiki/Rele>>. Päivitetty 6.8.2016. Luettu 3.4.2017.

Simulointi. 2016. Verkkodokumentti. Wikipedia. <<https://fi.wikipedia.org/wiki/Simulointi>>. Päivitetty 19.6.2016. Luettu 3.4.2017.

Transistori. 2017. Verkkodokumentti. Wikipedia. <<https://fi.wikipedia.org/wiki/Transistori>>. Päivitetty 18.2.2017. Luettu 3.4.2017.

Transistors. 2012. Verkkodokumentti. Salvius. <<http://blog.salvius.org/2012/07/transistors.html>>. Luettu 6.5.2017.

What is Arduino. 2017. Verkkodokumentti. Arduino AG. <<https://www.arduino.cc/en/Guide/Introduction>>. Luettu 3.4.2017.